

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Mobilná webová aplikácia pre
plánovanie ciest**

Travel Guide Mobile Web Application

Zadání bakalářské práce

Student:

Ján Baranec

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Mobilní webová aplikace pro plánování cest
Travel Guide Mobile Web Application

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je vytvořit mobilní webovou aplikaci pro vizualizaci dat z již existující aplikace pro plánování cest. Aplikace bude plně funkční i bez internetového připojení.

Aplikace bude dále obsahovat:

1. Časový harmonogram s různými typy pohledů.
2. Interaktivní mapu míst.
3. Navigaci mezi místy.
4. Poznámky.
5. Předpověď počasí pro jednotlivá místa.

Hlavní body zadání:

1. Přehled existujících řešení.
2. Seznámení s problematikou vývoje mobilních webových aplikací a mobilního webu.
3. Návrh a implementace aplikace.
4. User experience testování.

Seznam doporučené odborné literatury:

- [1] WEYL, Estelle. Mobile HTML5. 1st ed. xxvi, 450 pages. ISBN 1449311415.
- [2] ROZENTALS, Nathan. Mastering TypeScript. Packt Publishing - ebooks Account, 2015. ISBN 1784399663.
- [3] MENDOZA, Adrian. Mobile user experience: patterns to make sense of it all. Online-Ausg. Amsterdam [u.a.]: Morgan Kaufmann, an imprint of Elsevier, 2013. ISBN 0124095143.

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne
pramene a publikácie, z ktorých som čerpal.

V Ostrave 13. júla 2017


.....

Za pomoc pri tvorbe tejto práce by som chcel poďakovať viacerým ľuďom. Moja vďaka patrí predovšetkým Ing. Jánovi Janouškovi za poskytnutie cenných rád a odborného dohľadu počas celej tvorby práce. Ďalej Bc. Lubomírovi Sokolovskému za ochotu a pomoc počas vývoja tejto práce. Vďaka tiež patrí Bc. Diane Sokolovskej za gramatickú korektúru práce. V neposlednom rade by som sa chcel poďakovať svojej rodine za ich podporu počas celého štúdia.

Abstrakt

Cieľom tejto práce je vytvorenie mobilnej webovej aplikácie pre plánovanie ciest. Aplikácia využíva vektorové geodáta z otvorených zdrojov OpenStreetMap a vizualizuje ich do formy interaktívnej mapy. Mapa dokáže zobrazovať tiež miesta a trasy, ktoré sú vyhľadávané pomocou aplikačných rozhraní tretích strán. Aplikácia poskytuje cestovné harmonogramy pre plánovanie návštev zvolených miest. Využívajú sa vnútorné úložiská webového prehliadača, čo umožňuje prácu aj bez internetového pripojenia. Pre tvorbu aplikácie sú esenciálne prvky HTML5 a CSS3 štandardov. Funkcionalita aplikácie je implementovaná v transpilovanom, objektovo-orientovanom jazyku TypeScript.

Kľúčové slová: mobilná webová aplikácia, mapa, vektorové geodáta, OSM, OpenStreetMap, route, IndexedDB, WebStorage, HTML5, CSS3, TypeScript

Abstract

The aim of this thesis is a development of mobile web application for planning trips. Application uses vector-based geodata from open source of OpenStreetMap and visualizes them into form of interactive map. Map can also display places and routes, which are searched through application interfaces of 3rd parties. Application provides trip schedules for planning of visiting defined places. It uses inner storages of web browser, which allows it to function without internet connection. For development of this application are essential elements of HTML5 and CSS3 standards. Functionality of application is implemented in transpiled, object-oriented language TypeScript.

Key Words: mobile web application, map, vector geodata, OSM, OpenStreetMap, route, IndexedDB, WebStorage, HTML5, CSS3, TypeScript

Obsah

Seznam použitých zkratk a symbolů	9
Zoznam obrázkov	10
Zoznam tabuliek	11
1 Úvod	13
2 Analýza existujících řešení	14
2.1 Význam analýzy	14
2.2 Existující řešení	14
2.3 Výber prvkov pre aplikáciu	15
2.4 Datový model mapy	16
2.5 Výber platformy	18
2.6 Výber vývojových nástrojov	19
3 Zoznámenie s problematikou vývoja mobilných webových aplikácií a mobilného webu	20
3.1 Mobilný web design	20
3.2 Rýchlosť	21
3.3 Displej	22
3.4 Ovládanie	23
3.5 Uživatelské rozhranie	24
3.6 Navigácia	26
4 Návrh a implementácia aplikácie	29
4.1 Návrh	29
4.2 Implementácia	31
5 User Experience testovanie	42
5.1 Testované zariadenia	42
5.2 Podpora v prehliadačoch	43
5.3 User experience	43
6 Záver	47
Literatura	48
Přílohy	50

Seznam použitých zkratk a symbolů

OSM	– OpenStreetMap
HTML5	– Hyper Text Markup Language, version 5
CSS3	– Cascade style sheets, version 3
AJAX	– Asynchronous JavaScript and XML
POI	– Place of Interest
CRUD	– Create, Read, Update, Delete
DOM	– Domain Object Model
UI	– User Interface
REST	– Representational State Transfer
SOAP	– Simple Object Access Protocol
API	– Application Programming Interface
GPS	– Global Positioning System
GeoJSON	– Geographic JavaScript Object Notation
TopoJSON	– Topology JavaScript Object Notation
CDN	– Content Delivery Network
CD	– Compact Disc

Zoznam obrázkov

1	Využitie jednotlivých druhov zariadení priemerným používateľom počas celého dňa	18
2	Priemer a medián času potrebného na načítanie stránky na jednotlivých platformách	21
3	Náročnosť dotyku jednotlivých zón podľa držania mobilného zariadenia	24
4	Navigačný panel	26
5	Panel štítkov	26
6	Hamburger Menu	27
7	Navigačné centrum	27
8	Diagram tried predstavujúci domovskú stránku	30
9	Rozdiel medzi poradím vrstiev a problém vykreslenia názvov	34
10	Dynamické načítanie obsahu pri posune mapy	37
11	Ilustrácia použitia rozhrania domovskej stránky užívateľom	43
12	Ilustrácia vyhľadávania miest na mape užívateľom	44
13	Ilustrácia vyhľadávania trás medzi užívateľom definovanými bodmi na mape . . .	45
14	Ilustrácia manipulácie s obsahom projektov užívateľom	45

Zoznam tabuliek

- 1 Tabuľka podporovaných technológií v jednotlivých verziách prehliadačov [36] . . . 42

Seznam výpisů zdrojového kódu

1	Implementácia posunu mapy do strán	35
2	Obmedzenie počtu vyžadovaných sugescí na základe zadaného vstupu	38

1 Úvod

Zmyslom tejto práce je tvorba aplikácie, ktorá poskytne používateľovi jednoduchú a efektívnu orientáciu a dovoľí mu naplánovať si svoje cestovanie vopred. Pre orientáciu v priestore a navigáciu existuje mnoho nástrojov. Medzi najzákladnejšie, dodnes využívané vo viacerých oblastiach, patrí mapa. Z významných dôvodov sa v dnešnej dobe používa práve jej digitálna forma. Preto práve tento druh mapy je jedným z esenciálnych prvkov tejto aplikácie. Popri mape aplikácia obsahuje aj dodatkové prvky, ktoré jej umožňujú plnohodnotné plánovanie výletov. Medzi tieto prvky patrí napr. rozhranie na tvorbu cestovných harmonogramov či výpočet optimalizovanej trasy medzi viacerými bodmi, ale aj niekoľkodňová predpoveď počasia.

Práca sa v prvej časti zaoberá porovnávaním existujúcich riešení aplikácii zameraných na plánovanie ciest. Z analýzy týchto riešení sú vybrané vhodné prvky, ktoré sú použité ako požiadavky pre cieľový produkt. Po špecifikácii týchto požiadavok sa následne zvolí datový model pre interaktívnu mapu. Po voľbe modelu mapy, autor pokračuje výberom cieľovej platformy, pre ktorú je vhodný produkt vyvíjať. Nasledne autor zvolí vhodné nástroje pre vývoj samotný. Je odôvodnený výber platformy, pričom sa zohľadňuje jej dostupnosť pre užívateľa a vlastnosti dôležité pre cieľový produkt.

Druhá časť práce sa zaoberá úskaliami a špecifickými problémami vývoja na vybranej cieľovej platforme pre aplikáciu. Sú tu vytýčené základné rozdiely medzi vývojom mobilných a desktopových aplikácií. Touto časťou je zakončená teoretická časť práce.

V tretej časti je popísaný návrh a samotná implementácia aplikácie. Pomocou diagramov sú ilustrované objekty aplikácie a väzby medzi nimi. Účel každého objektu je detailne popísaný. Následne sú v stručnosti objasnené problémy spojené s vývojom našej aplikácie.

Štvrtá časť práce opisuje testovanie a použitie cieľového produktu s užívateľského pohľadu. Väčšia časť tejto kapitoly sa zameriava práve na ilustrovanie ovládania užívateľského rozhrania aplikácie. Okrem toho sú tiež popísané kladné a záporné stránky rozhrania, pričom sú uvedené aj návrhy na vylepšenie.

V poslednej, záverečnej časti sú kriticky zhodnotené celkové úspechy a neúspechy práce s prípadnými nápadmi pre ďalší rozvoj aplikácie.

2 Analýza existujúcich riešení

2.1 Význam analýzy

Na trhu nájdeme mnoho variácií produktov, ktoré sa snažia splniť ten istý účel vo vlastnom, jedinečnom prevedení. Vývoj týchto produktov je zväčša opakujúci sa proces, v mnohých ohľadoch rozdielny len v jeho špecifických požiadavkách. Obyčajne sa tieto produkty líšia drobnými vylepšeniami na funkcionalite, prípadne rozdielnou implementáciou užívateľského rozhrania. Mnohokrát však tieto aplikácie majú množstvo spoločných a overených prvkov, ktoré jej umožňujú úspešne plniť jej účel.

Identifikovaním a použitím takýchto overených riešení je možné predísť množstvu v minulosti riešených problémov. Jedná sa predovšetkým o zjednodušenie celkovej náročnosti implementácie produktu, nakoľko nie je nutné vymýšľať tieto prvky od základu. Prispievajú aj k intuitívnejšiemu ovládaniu aplikácie pre samotného užívateľa, nakoľko je s týmito prvkami oboznámený z predošlých skúseností s podobnými aplikáciami. Výsledkom je v konečnom dôsledku znížený čas potrebný na vývoj aplikácie a tiež čas potrebný na osvojenie si jej rozhrania užívateľom.

Využitím overených riešení sa nesporne zvyšuje efektívnosť i úspešnosť výsledného produktu, a preto sa táto časť zameriava práve na ich vyhľadanie.

2.2 Existujúce riešenia

2.2.1 Roadtrippers

Roadtrippers je webovou aplikáciou s podporou pre mobilné platformy [1]. Podľa oficiálnych zdrojov má viac ako 5.5 milióna uložených rozplánovaných ciest [2]. Pri takomto využití služieb sa dá považovať aplikácia za pomerne úspešnú. Roadtrippers ponúka užívateľovi rozhranie pre vyhľadávanie konkrétnych miest alebo plánovanie ciest medzi nimi. Toto rozhranie existuje v dvoch funkčne totožných formách - webová aplikácia a mobilná aplikácia.

K orientácii aplikácia poskytuje interaktívnu mapu zeme, ktorá tvorí prevažnú väčšinu obsahu aplikácie. Pre užívateľa sú vizualizované miesta, trasy a tiež body záujmu(POI)¹. Napriek množstvu takýchto bodov je mapa prehľadná, čo ovplyvňuje hlavne postranné menu. To odobremňuje mapu od zobrazovania množstva detailnejších informácií, ktoré by ju zbytočne zneprehľadňovali. Poskytuje tiež rozhranie pre vyhľadávanie miest a trás na mape a taktiež filtráciu bodov záujmu podľa kategórii. V rámci trás zobrazuje čas potrebný k presunu ako aj vzdialenosť medzi bodmi. V postrannom menu je tiež možnosť okamžitej rezervácie vo užívateľom vybranom bode záujmu. K doplnkom aplikácie patria aj kolekcie predpripravených výletov, ktoré je možné rámci cestovania rezervovať a navštíviť. Napriek pokročilým funkciám aplikácia však nepodporuje predpoveď počasia, v rámci cestovania pomerne významnú vec.

¹POI - Places of Interest - sú napr. reštaurácia, park, divadlo, nákupné strediská atď.

Hlavnými prvkami aplikácie sú :

- Interaktívna mapa
- Postranné menu
- Rezervácie
- Predpripravené plány

2.2.2 Sygic Travel

Sygic je spoločnosť ktorá sa zameriava na vývoj navigačných aplikácií. Jej aplikácie využíva viac ako 150 000 000 vodičov, čo je hlavný dôvod, pre ktorý bola vybraná k analýze. [3]. Spomedzi množstva jej úspešných produktov bola zvolená aplikácia Sygic Travel (pôvodná Tripomatic) [4]. Tá je zameraná na plánovanie ciest a výletov. Existuje ako webová aplikácia a tiež aj ako mobilná aplikácia.

Sygic Travel poskytuje užívateľovi rozhranie pre plánovanie ciest v rámci celého sveta. Rovnako ako aj aplikácia Roadtrippers, obsahuje Sygic Travel interaktívnu mapu zeme. Mapa dokáže znázorniť množstvo bodov záujmu, ktoré je možno filtrovať a medzi ktorými si môže užívateľ podľa svojej vôle vyberať. Výberom takéhoto bodu aplikácia zobrazí detaily samotného bodu v menu nachádzajúcom sa pri mape. V rámci zvoleného miesta na mape aplikácia tiež navrhuje v postrannom menu užívateľovi možnosť rezervácie pobytu, prenájmu aut či Hop on Hop off služieb. Zobrazujú sa tu tiež detaily trasy, ako je dĺžka a čas potrebný na presun zvoleným dopravným prostriedkom. Mobilná verzia aplikácie je schopná zobrazovať celosvetovú mapu v offline režime rovnako ako aj výpočet pešej navigácie. Táto verzia má tiež rozhranie na predpoveď počasia pre danú oblasť.

Hlavnými prvkami aplikácie sú :

- Interaktívna mapa
- Postranné menu
- Rezervácie
- Predpoveď počasia
- Možnosť práce v režime offline

2.3 Výber prvkov pre aplikáciu

Z analyzovaných vzorových aplikácií je patrne vidieť, že obsahujú množstvo spoločných prvkov. Hlavným a najdôležitejším prvkom je interaktívna mapa, nakoľko poskytuje užívateľovi základné rozhranie na orientáciu v priestore. Práve preto sa bude v tejto aplikácii implementovať. Užívateľ si na nej bude môcť zvoliť trasu medzi viacerými bodmi. Ďalším pomerne významným prvkom vybraným pre túto aplikáciu je postranné menu. To bude poskytovať dodatočné ovládanie mapy a bude zobrazovať detailnejšie informácie zobrazených miest. Tým sa docieli zjednodušenie

orientácie v mape. Aplikácia bude obsahovať tiež niekoľkodňovú predpoveď počasia pre zvolenú lokáciu. Na základe tejto predpovede sa môže užívateľ lepšie rozhodovať pri svojom plánovaní trasy. Táto predpoveď bude umiestnená v postrannom menu spolu s ďalšími detailami zvoleného miesta. Aplikácia bude implementovaná tak, aby bola schopná v rámci možností pracovať aj v offline režime.

2.4 Datový model mapy

V rámci existujúcich riešení je možné mapu vykresľovať z dvoch základných datových modelov - rástrového a vektorového [5].

Rástrový datový model - informácie o priestore sú spravidla kódované do pixelov obrázkov.

Tieto dáta sú delené do rovnako veľkých blokov umiestnených v mriežke. Toto kódovanie prebieha dopredne na strane servera, ktorý tieto dáta poskytuje. V prípade že by sme chceli takéto dáta využiť, server nám pošle hotový obrázok vyžiadanej oblasti.

Výhody

- jednoduchý vstupný formát dát pre mapu - hotové obrázky prijímané zo serveru
- menšie výpočetné nároky na zobrazenie mapy
- jednoduchší na implementáciu

Nevýhody

- vstupné data sú fixnej veľkosti - nemožnosť zmeny veľkosti vykreslenia blokov
- vstupné data môžu byť obrovskej veľkosti - náročne na prenos sieťou
- možná strata informácií o priestore
- okrem samotného obrázku nemáme ďalšie informácie o topológii oblasti
- zložitá implementácia prekrývania vrstiev mapy

Vektorový datový model - informácie o priestore sú ukladané do štrukturovaných objektov, ktoré sa skladajú z geometrických útvarov (čiary, body, polygóny) a atribútov. Geometrické útvary majú za úlohu popisovať tvar a pozície objektov reálneho sveta na mape (napr. čiara je cesta) a atribúty opisujú objekt samotný (napr. názov ulice, výška budovy). Oproti rástrovému modelu sa vizuálna forma týchto dát rozhoduje až na strane klienta.

Výhody

- jednoduché škálovanie veľkosti zobrazovaných dát
- jednoduchšia implementácia prekrývania vrstiev
- menšie veľkosti vstupných dát ako rástrové data
- kompletný opis topológie oblasti
- kompatibilnejší v rámci ukladania do relačných databáz
- umožňuje analýzu topológie (výpočet trás atď.)

Nevýhody

- väčšie výpočetné nároky na zobrazenie mapy
- komplexné dátá sú ťažšie na manipuláciu
- celkovo náročnejšia implementácia

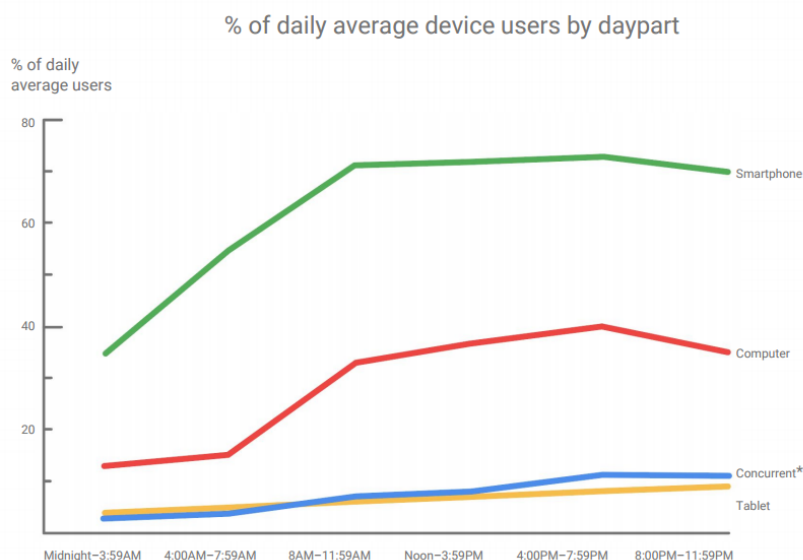
Ráastrový model nám oproti vektorovému ponúka ľahšiu manipuláciu dát a celkovo jednoduchšiu implementáciu mapy. Jeho znížené výpočetné nároky sú pre výkonovo slabšie mobilné platformy pomerne vítanou vlastnosťou. Zobrazovanie vektorových dát je síce výpočtovno náročnejšou operáciou, no vďaka využitiu cachovania vykreslených blokov je možné redukovať túto výpočtovú náročnosť len na počiatočné vykreslenie. Vektorový model navyše znižuje množstvo dát prenesených cez sieť. To je spôsobené nielen menšiou veľkosťou datových blokov, ale aj možnosťou ich ukládania do indexovanej databázy moderných prehliadačov. To pre tento model znamená, že každý nezmenený blok sa sťahuje len jedenkrát. Vektorový model nám oproti ráastrovému tiež umožňuje bezstrátovo škálovať zobrazenie datových blokov. Okrem toho navyše poskytuje aj kompletný opis topológie danej oblasti. To sa dá využiť pri analyzovaní či vyhľadávaní v datových blokoch vybraných území.

Na základne vyššie zmienených faktov bol pre mapu zvolený vektorový model, nakoľko jediným neovlivniteľným negatívnym faktorom je jeho náročnejšia implementácia. Zvyšné negatíva tohto modelu sa dajú takmer v celom rozsahu eliminovať využitím dnes dostupných prostriedkov ako je cachovanie a ukladanie do databázy. Navyše nám poskytuje oveľa lepšie možnosti v rámci jeho vykresľovania do mapy.

2.5 Výber platformy

Účelom našej aplikácie je užívateľovi poskytnúť rozhranie pre navigáciu a plánovanie cesty. V rámci vývoja aplikácie je však nutné rozhodnúť aká platforma je pre tento účel najvhodnejšia.

Z praktického hľadiska sa dá predpokladať, že užívateľ aplikácie si počas svojho cestovania bude chcieť overiť, či cestuje podľa určeného plánu, prípadne si tento plán môže chcieť pozmeniť. Nato, aby mohol niečo také uskutočniť, potrebuje zariadenie, ktoré mu bude plne prístupné aj počas jeho cestovania. Pre tento účel mu pravdepodobne najvhodnejšie poslúžia mobilné zariadenia (obrázok 1). Tie sú významne menšej veľkosti ako platformy desktopové, čo ich robí jednoducho prenositeľnými a tým pádom aj prístupnejšími počas celého dňa.



Obr. 1: Využitie jednotlivých druhov zariadení priemerným používateľom počas celého dňa [6]

V poslednej dekáde sa podľa štatistických prieskumov každoročne tiež zvyšuje počet užívateľov vlastniacich takéto zariadenie [7]. Podľa Global Web Index, ktorý v treťom kvartále roka 2016 vykonal prieskum využitia mobilných zariadení vo viac ako 40 krajinách po celom svete.

"Smartfóny momentálne prebehli PC/Laptopy ako najviac bežne vlastnené zariadenia - 91% opýtaných osobne vlastní aspoň jeden" [8]

"Digitálny konzumenti momentálne identifikujú mobilné telefóny ako ich najdôležitejšie zariadenie pre prístup k internetu" [8]

V rámci prehliadania internetového obsahu sa teda prevládajúcimi aktuálne stávajú mobilné zariadenia [9]. Podľa odborných predikcií sa toto množstvo má v najbližších rokoch naďalej zvyšovať [10].

Na základe vyššie zmienených faktov boli mobilné platformy prehodnotené a zvolené ako primárna cieľová platforma pre túto aplikáciu.

2.6 Výber vývojových nástrojov

Prezatiaľ boli zvolené samotné ciele aplikácie i vhodné cieľové platformy pre ktoré sa táto aplikácia bude vyvíjať. Pre dosiahnutie nami zvoleného cieľa je však tiež vhodné vybrať i správne nástroje.

Značkovací jazyk HTML patrí k základným technológiám potrebným pre vývoj akýchkoľvek webových stránok. Účelom tohto jazyka je definícia základnej štruktúry a obsahu stránok aplikácii. Konkrétne pre našu aplikáciu bol zvolený HTML5, v ktorom sa poprvýkrát objavuje `<canvas>` element. Význam tohoto elementu spočíva vo vykresľovaní grafiky na určenú plochu. Tým sa stáva ideálnym pre tvorbu interaktívnej mapy.

Jazyk CSS je štandardným jazykom pre štylovanie obsahu webových aplikácií a pre tento účel bude použitý aj pri vývoji našej aplikácie. Využitá bude najnovšia verzia 3.

JavaScript je dynamický, netyповaný, interpretovaný skriptovací jazyk, ktorý je používaný na implementáciu dynamického obsahu webových aplikácií. Jedná sa predovšetkým o spracovanie udalostí, animácie a komunikáciu cez sieť. Pre vývoj tejto aplikácie bude však použitý transpilovaný jazyk TypeScript ².

"TypeScript je silne typový, objektovo orientovaný, kompilovaný jazyk, ktorý umožňuje programátorovi používať koncepty a základné idey osvedčených objektovo-orientovaných jazykov - v JavaScripte. TypeScript generuje JavaScript - dodržiavajúc pritom koncepty silne typových a objektovo-orientovaných jazykov - ale zároveň však zostáva len čistým JavaScriptom [11]."

TypeScript teda umožní implementovať aplikáciu s použitím silného typovania a objektovo-orientovaných konceptov. To nám umožní vyššiu udržateľnosť a prehľadnosť kódu v rámci komplexity našej aplikácie.

V jazyku JavaScript existuje množstvo voľne dostupných knižníc, ktoré napomáhajú k jednoduchšiemu vývoju dynamických webových aplikácií akou je aj táto. Jednou s týchto knižníc je aj jQuery.

"jQuery je rýchla, malá a na doplnky bohatá JavaScriptová knižnica. Robí veci ako manipuláciu a prechod HTML dokumentami, spracovanie udalostí, animácie a Ajax oveľa jednoduchšími spolu s API, ktoré je jednoduché na použitie a pracuje naprieč množstvom prehliadačov [12]."

Knižnica jQuery bola vybraná pre jej schopnosť zjednodušiť manipuláciu so základnými prvkami stránok a tým znížiť celkovú náročnosť tvorby tejto aplikácie.

²transpilovaný/transkompilovaný jazyk - prekladá kód jazyka do inej verzie toho istého jazyka. Táto verzia býva v určitom zmysle výhodnejšia ako pôvodná (napr. syntactic sugar)

3 Zoznámenie s problematikou vývoja mobilných webových aplikácií a mobilného webu

Vývoj webových aplikácií pre vybranú platformu, či už desktopovú alebo mobilnú, je spojený s množstvom výhod i nevýhod. Každá táto platforma je totiž jedinečná a prináša do vývoja špecifické možnosti, ale aj obmedzenia. Pre úspešný vývoj mobilnej webovej aplikácie je nutná znalosť zásadných rozdielov medzi týmito platformami. Hoci je táto problematika oveľa širšia než rozsah tejto práce, pár nasledujúcich podkapitol sa bude snažiť priblížiť čitateľovi aspoň najdôležitejšie časti tejto problematiky.

3.1 Mobilný web design

Obvykle existuje viacero spôsobov, ako dosiahnuť rovnaký cieľ, pričom vývoj mobilného webu nieje výnimkou. Každý spôsob vnáša do vývoja špecifické prínosy. Je potrebné si ich uvedomiť a na základe nich rozhodnúť, akým spôsobom sa vývoj bude odohrávať. V podstate existujú tri druhy vývoja mobilných webových aplikácií, a to sú:

Oddelená mobilná doména - pre mobilnú a desktopovú verziu webu existujú špeciálne domény. Takto koexistujú súčasne dve verzie rovnakého webu, pričom každá je dimenzovaná na inú platformu. Pri pokuse užívateľa navštíviť jednu s týchto verzií, user-agent³ zistí a pošle serveru informácie o platforme, ktorá sa snaží web domény prehliadať. Na základe nich server presmeruje užívateľa na odpovedajúcu doménu [13].

Adaptívny web design - pre web existuje jediná doména. Tá však obsahuje stránky dimenzované pre všetky rozlíšenia oboch platforiem. Pri návšteve webu sa taktiež využíva user-agent, ktorý na server zasiela informácie o platforme a rozlíšení zariadenia. Základe týchto informácií potom server užívateľovi pošle webové stránky, ktoré sú dimenzované na jeho platformu a rozlíšenie [13].

Responzívny web design - pre web existuje jediná doména. Tá obsahuje jedinou verziu stránok, ktorá sa používa pre všetky rozlíšenia a platformy. Charakteristickým rozdielom tohto designu je, že pre štylovanie prvkov stránky používajú nie absolútne, ale relatívne rozmery. Prípadná nutnosť preusporiadania obsahu je detekovaná a riešená pomocou CSS3 Media Queries. Pre vývoj to znamená, že sa obsah vždy zobrazuje adekvátne k rozlíšeniu, pre ktoré sa má zobrazíť [13].

Responzívny web design býva z dobrých dôvodov jasná voľba medzi týmito tromi variantami. Oddelená mobilná doména je síce počiatočne jednoduchšia na implementáciu, ale pri väčších weboch je nevýhodné a komplikované udržiavať dve verzie rovnakého webu. Tie nielenže zaberajú 2x toľko priestoru, ale presmerovanie na tieto domény býva náchylné na chyby. Adaptívny design tiež ponúka jednoduchšiu počiatočnú implementáciu v rámci jednej domény, avšak stále

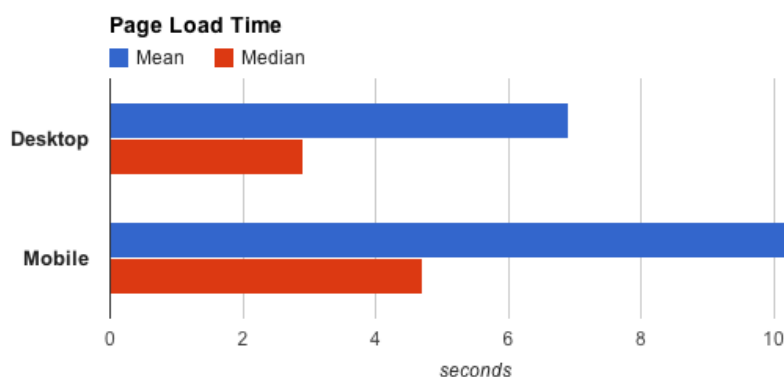
³User-agent je software, ktorý v zastúpení za užívateľa vykonáva definovanú činnosť.

s dvojitém množstvom obsahu, ktoré je taktiež náchylné na chyby a je ťažšie udržiateľné. Responzívny design je schopný meniť svoj obsah vzhľadom k prostrediu, v ktorom je zobrazovaný. Jeho počiatočná implementácia je náročnejšia, no vyžaduje jedinú doménu a zaberá len zlomok priestoru v porovnaní s ostatnými designami [13] [14].

Pre našu aplikáciu bol na základe vyššie zmienených dôvodov zvolený Responzívny web design. Vybraný bol predovšetkým vďaka jeho jednoduchšej udržiateľnosti, priestorovej nenáročnosti a minimálnej réžii pre jeho chod.

3.2 Rýchlosť

Nakoľko mobilné platformy sú oproti desktopovým značne menšej veľkosti, prirodzene teda oplývajú menším výpočtovým výkonom. Pre mobilné webové aplikácie to znamená, že vyžadujú oveľa viac času na spracovanie obsahu stránky a tiež na jej celkový chod. Samozrejme, je nutné brať do úvahy aj povahu aplikácie, nakoľko nie každá má rovnaký účel, prípadne typ interakcií v nej. Z obrázku 2 je patrne vidieť, že čas potrebný na načítanie stránky mobilnými platformami oproti desktopovým je vyšší.



Obr. 2: Priemer a medián času potrebného na načítanie stránky na jednotlivých platformách [15]

V rámci rýchlosti načítavania stránok, je tiež nutné si uvedomiť, že celková rýchlosť a prístupnosť do siete na mobilných zariadeniach nemusí byť vždy zaručená. Fluktuácia prenosovej rýchlosti je pri takýchto zariadeniach pomerne častým problémom.

Prieskum vytvorený Google Analytics ukazuje že :

"53% stránok je užívateľom opustených, ak ich načítanie trvá dlhšie ako 3 sekundy [16]."

"50% užívateľov očakáva, že ich webová stránka sa načíta do menej ako 2 sekúnd [16]."

Práve pre tieto dôvody by malo byť snahou každého vývojára mobilných webových aplikácií, redukovat množstvo potrebného času na načítanie a chod jeho aplikácie.

K tomuto účelu je možné využiť viacero prostriedkov. Medzi najvýznamnejšie, na ktoré je vhodné sa sústrediť, patrí redukcia množstva a veľkosti obrázkov, videí a audio súborov. Tie sú nielen obzvlášť náročné na prenos cez sieť, ale výpočtovo náročná je aj ich úprava. Ďalej sa jedná o optimalizáciu komunikácie cez sieť v podobe minimalizovania veľkosti a množstva prenášaných súborov [17]. Cachovanie dát je vysoko efektívnym nástrojom, ktorý umožňuje zníženie prenosu cez sieť, ale aj minimalizáciu spracovania dát. V prípade, že aplikácia nevyhnutne potrebuje využívať väčší objem súborov, doporučuje sa ich poskytovať skrz službu Content Delivery Network (CDN).

"Content Delivery Network je kolaboratívna kolekcia sieťových prvkov rozširujúcich Internet, kde obsah je replikovaný na viacerých zrkadlených webových serveroch, za účelom vykonať transparentné a efektívne doručenie obsahu koncovým užívateľom [18]."

Výhoda ukladania zdrojov do CDN je ich neustála dostupnosť a rýchla prístupnosť. Zdroje sú totiž vždy poskytované od najbližšieho servera vzhľadom k užívateľovej pozícii. Okrem vyššie uvedených praktík, je rovnako dôležité aj optimalizovať dynamické prvky stránky, ako napríklad počet animácií a celková optimalizácia interakcie medzi prvkami. Odporúčané je tiež minimalizovať počet prvkov v štruktúre stránky, rovnako ako aj počet štýlovacích pravidiel [17].

Naša aplikácia obsahuje mapu, ktorá je výpočetne-náročným prvkom. Navyše, táto mapa sama o sebe vyžaduje pre jej funkciu relatívne veľký objem dát. Práve preto bol kladený významný dôraz nielen na rýchlosť načítania stránky, ale taktiež aj na celkovú rýchlosť aplikácie. Metóda cachovania obsahu je využívaná ako pre dáta mapy, tak pre obsah užívateľského rozhrania aplikácie. Rozhranie je schopné cachovať užívateľom vybrané projekty, harmonogramy, ich miesta a tiež poslednú pozíciu zobrazenú mapou. Mapa zase cachuje obsiahle vektorové geodáta, ktoré má za úlohu vizualizovať. Pre zefektívnenie komunikácie medzi API tretích strán, ktoré našej aplikácii poskytujú dáta o počasí, sa využíva tzv. throttling ⁴. Multimediálny obsah stránky je minimalizovaný natoľko, aby zbytočne nezatažoval prenos po sieti a chod aplikácie. Tento obsah tvoria predovšetkým fotografie, ktoré sú prístupné aj pomocou služieb CDN.

3.3 Displej

Ďalšou dôležitou stránkou mobilných platforiem je veľkosť ich displeja. V porovnaní s desktopovými displejmi sú pochopiteľne menšie, nie len však fyzicky, ale aj rozlíšením. Menší displej nedovoľuje užívateľovi zároveň vidieť také množstvo obsahu, aké by mohol na desktopovom zariadení. Častokrát si teda musí užívateľ posúvať stránku v určitom smere, aby bol schopný vidieť aj jej zvyšný obsah [17][19].

Vývojár teda musí dbať, aby užívateľské rozhranie prezentovalo podstatné informácie, a to čo najprehľadnejšie. Dobrým zvykom býva umiestňovať najdôležitejšie informácie na začiatok

⁴throttling (preložené priškrcovanie) - je technika, ktorou sa obmedzuje množstvo vykonanej komunikácie v určenom intervale

stránky, tak aby ich užívateľ nemal možnosť prehliadnúť. Prípadne sa odporúča implementovať navigačný prvok, na základe ktorého sa môže užívateľ orientovať vo zvyšnom obsahu stránky. Ďalšou, nemenej dôležitou praktikou, je na relatívne malé oblasti neumiestňovať veľké množstvo rozličných prvkov. To by mohlo nielen zbytočne skomplikovať celkovú orientáciu v UI, ale vzhľadom k povahe dotykového ovládania, tiež aj znehodnotiť kvalitu jeho interakcie. K zvýšeniu celkovej prehľadnosti sa tiež doporučuje adaptívne k rozlíšeniu displeja meniť veľkosť fontov, čo užívateľovi umožňuje lepšie prečítať text stránky [17][19].

Naša aplikácia je implementovaná vzhľadom k vyššie zmieneným faktorom. Do vrchných častí stránok sú umiestnené najdôležitejšie informácie. Jedná sa predovšetkým o tie, na základe ktorých je schopný sa užívateľ orientovať vo zvyšku obsahu aplikácie (napr. účel stránky a bližšie informácie o nej). Pre navigáciu medzi týmito stránkami je do aplikácie umiestnený bočný panel. Ten je užívateľovi prístupný v každom mieste aplikácie. Autorovou snahou bolo tiež minimalizovať počet prvkov UI. Ich prílišná početnosť je pre užívateľa rušivým faktorom.

3.4 Ovládanie

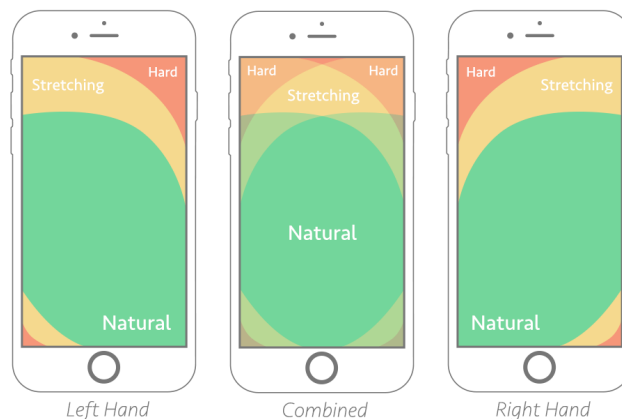
Významným faktorom v rámci využitia mobilných zariadení je aj jeho ovládanie. Narozdiel od desktopových platforiem, je v drvivej väčšine užívateľ odkázaný na ovládanie dotykovým displejom. Tento spôsob ovládania má oproti perifériám, akými sú myš a klávesnica množstvo rozdielov.

Jedná sa predovšetkým o zníženú presnosť a rýchlosť ovládania. V tomto ohľade sa ovládanie dotykom aj pri sebväčšej snahe, nemôže rovnať myši. Je pomerne bežnou situáciou, že dotyk sa buď vyberie neželaný alebo žiaden prvok. Navyše celá interakcia sa deje častokrát pomalšie. Takáto náchylnosť k chybovosti obzvlášť narastá v prípade, keď užívateľské rozhranie obsahuje väčšie množstvo menších prvkov. Typickým príkladom je práca s väčším objemom textového obsahu. Pre užívateľa je teda celkovo frustrujúce, pokiaľ je v užívateľskom rozhraní veľa drobných prvkov, a keď od neho aplikácia vyžaduje vysoký stupeň interakcie [19].

Špecifickým, výhodným a zaužívaným druhom ovládania dotykových displejov sú gestá. Pomocou imitácií pohybov ťahania, roztahovania, sťahovania a mnohých iných, je užívateľ schopný odďaľovať, približovať či ináč interagovať s obsahom stránky [17]. Implementácia gest v mobilnej webovej aplikácii síce nieje nutnosťou, ale ponúka možnosť užívateľovi poskytnúť originálny a zaujímavý spôsob ovládania.

Podstatné je si tiež uvedomiť, že vzhľadom k uchopeniu mobilného zariadenia v ruke, je na niektoré oblasti jeho displeja oveľa ťažšie dosiahnúť prstami (Obrázok 3). Je preto vhodné rozmiestňovať najpoužívanejšie prvky do najjednoduchšie prístupných zón na displeji [20].

Je možné teda povedať, že aplikácie ovládané dotykom majú svoje špecifické výhody, ale i nevýhody. Množstvo nevýhod sa ale v rámci optimalizovaného návrhu dá odbúrať. Všeobecným predpokladom úspešnej aplikácie, ktorá využíva dotyk ako formu ovládania je, aby vyžadovala čo najmenej nutnej užívateľskej interakcie. To z časti eliminuje problém pomalého a nepresného ovládania dotykom. Vzhľadom k ovládaniu, obsah a štruktúra aplikácie by zároveň mali byť



Obr. 3: Náročnosť dotyku jednotlivých zón podľa držania mobilného zariadenia [20]

postavené, pokiaľ možno, čo najjednoduchšie. Rozhrania by mali obsahovať rozumné množstvo prvkov, s veľkosťou optimalizovanou pre ovládanie dotykom. Predíde sa tak chybovosti pri výbere prvkov [19]. Okrem toho, medzi ďalšie efektívne spôsoby ako vylepšiť ovládanie aplikácie patrí napr. minimalizácia počtu vstupných prvkov. Pre vyhľadávacie prvky sa tiež odporúča implementácia sugescie. Vďaka nej totiž užívateľovi odpadáva nutnosť zadávania množstva dát ručne, čím sa znižuje chybovosť a zrýchľuje celková interakcia s aplikáciou [21].

Do našej aplikácie bola implementovaná väčšina spomenutých princípov. Konkrétnejšie sa jedná predovšetkým o optimalizáciu prvkov pre ich nenáročné ovládanie pomocou dotyku. Do interaktívnej mapy bolo implementované ovládanie gestami. Vďaka nim sa môže užívateľ intuitívnym spôsobom na mape buď približovať, oddalovať alebo ľubovoľne posúvať jej obsah. Pomocou gest je tiež možné pristupovať k hlavnému navigačnému prvku aplikácie. Rozhranie je celkovo stavané s dôrazom na jednoduchosť, to platí ako pre štruktúru obsahu, tak aj pre množstvo potrebnej interakcie zo strany užívateľa.

3.5 Uživatelské rozhranie

V porovnaní s desktopovým platformami, je tvorba správneho užívateľského rozhrania pre mobilné platformy vo viacerých smeroch komplikovanejšia. Stojí za tým množstvo faktorov spojených s prirodzenou povahou mobilných zariadení.

Prvým z nich je už spomínaný displej. Viditeľná plocha mobilných stránok býva zhruba 3-krát menšia ako je tomu u desktopových zariadení. S tým sa spája množstvo komplikácií, ktoré musí vývojár ošetriť už pri návrhu. Prvky je totiž nutné navrhnuť tak, aby užívateľ bol schopný jednoduchej orientácie v celom obsahu stránky, aj napriek tomu, že vidí len jeho časť. Z prieskumu spoločnosti Google, v ktorom bolo uvedené, že:

"61% užívateľov povedalo, že ak na stránke nenašli hneď to čo hľadajú, rýchlo sa presunuli na inú stránku." [22]

"79% ľudí, ktorým sa nepáči to, čo nájdú na jednej stránke sa vráti, a začne hľadať inú stránku." [22]

Práve kvôli týmto dôvodom je nutné, aby sa užívateľ mal možnosť orientovať v aplikácii čo najrýchlejšie. Preto sa najdôležitejšie prvky takmer vždy umiestňujú na začiatok stránky, odkiaľ ich užívateľ môže aj hneď začať využívať. V rámci zachovania prehľadnosti je však nutné dbať nato, aby sa v jednej oblasti nevyskytovalo prvkov príliš veľa.

Aplikácia by mala tiež ponúkať sebestačné užívateľské rozhranie. To konkrétnejšie znamená, že užívateľ by v ňom mal nájsť všetky potrebné informácie pre plnohodnotný chod aplikácie. Dohľadávanie informácií, ktoré aplikácia potrebuje, ale neposkytuje, dokáže byť vyčerpávajúce, časovo náročné a často odradzujúce pre užívateľa. Nehovoriac o tom, že tieto informácie by mal po častiach kopírovať z iných stránok, prípadne si ich niekde bokom zapisovať. To by bolo značne komplikované, nakoľko užívateľ ani nemá možnosť používať na mobilnom prehliadači viac ako jedno okno webového prehliadača zároveň [19].

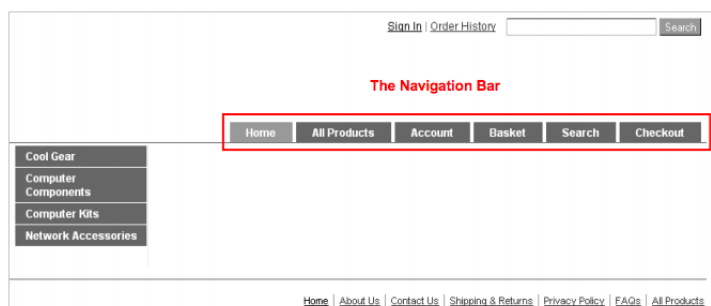
Užívateľské rozhranie by malo byť navrhnuté proti častým prerušeniam. Je dôležité si uvedomiť, že pre užívateľov mobilných zariadení sú charakteristické krátke relácie. Užívateľia totiž často prehliadajú mobilný web v rôznych situáciách a to počas celého dňa. Častým výsledkom takéhoto prehliadania je náchylnosť k prerušeniu jej používania. Užívateľ týmto spôsobom môže častokrát prísť o dočasné údaje prehliadania. Efektívnym riešením ako eliminovať tento problém, je automaticky a pravidelne ukladať medziúdaje prehliadania, alebo užívateľovi aspoň poskytnúť možnosť ich uložiť manuálne. Útržkovitá pozornosť venovaná aplikácii, ktorá je spojená s častými prerušeniami, by mala byť tiež ošetrovaná. Pre tento účel je nutné, aby dokázala pracovať samostatne vo väčších logických blokoch, bez potreby neustálej interakcie s užívateľom [19].

Užívateľské rozhranie bolo navrhované s ohľadom na vyššie uvedené faktory. Najhlavnejšie prvky boli prioritne umiestnené na vrch stránky, čím sa efektívne zabraňuje ich prehliadnutiu. Jedná sa predovšetkým o mapu a jej kontrolný panel. Pre jednoduchú orientáciu bolo implementované navigačné menu, ktoré umožňuje navigáciu v aplikácii. Toto menu je prístupné pre užívateľa na akomkoľvek mieste aplikácie. Aplikácia je schopná taktiež ukladať stav, konkrétnejšie užívateľom zvolený projekt. V rámci možností bola realizovaná sebestačnosť aplikácie. Užívateľ by mal byť schopný v nej nájsť všetky potrebné informácie pre jej chod.

3.6 Navigácia

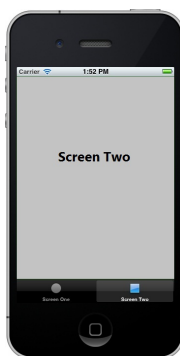
Navigácia zastáva jednu s najdôležitejších úloh v rámci mobilných webov. V kontraste s desktopovými webmi, mobilné oplývajú menšou viditeľnou plochou. Užívateľ tak nemá možnosť si na prvý pohľad utvoriť mentálny model celej stránky. Na tento účel slúžia navigačné prvky. Tie by mali byť pre užívateľa ľahko objaviteľné a jednoducho prístupné. Pre implementáciu takýchto navigačných prvkov existuje množstvo overených vzorov. Každý z týchto vzorov má svoje špecifické výhody a nevýhody v rámci celkového designu webu [23]. Medzi najpoužívanéjšie navigačné prvky patria:

Navigačný panel - je horizontálnym zoznamom textových položiek. Tento prvok býva umiestnený na vrchu, relatívne k celej stránke. Je vhodný pre menšie množstvo možností, nakoľko stránky bývajú tvorené viac do dĺžky ako šírky (Obrázok 4) [23].



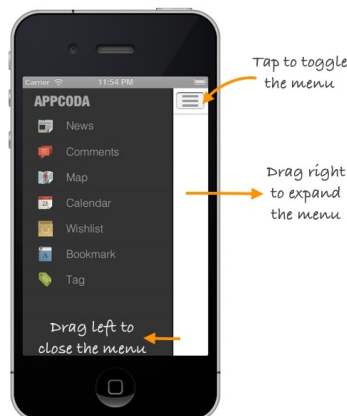
Obr. 4: Navigačný panel
[24]

Panel štítkov - je taktiež horizontálnym zoznamom, no namiesto textu obsahuje štítky. Umiestnený býva buď na vrchnej alebo spodnej časti stránky. Rovnako ako navigačný panel, nedokáže efektívne zobrazovať väčšie množstvo položiek (optimálny počet je 5 a menej). Narozdiel od neho, panel štítkov má umiestnenie relatívne k viditeľnej ploche (Obrázok 5) [23] [25].



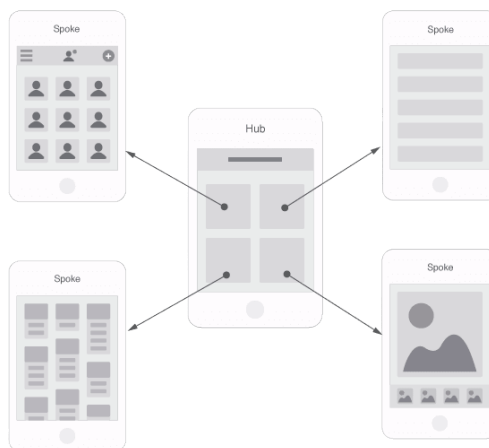
Obr. 5: Panel štítkov
[26]

Hamburger Menu - jedná sa o vertikálny zoznam položiek. Toto menu býva obvykle skryté, ktoré sa po vyžiadaní užívateľovi zobrazí. Zvyčajne býva ukryté v jednej zo strán užívateľského rozhrania, odkiaľ sa aj vysúva pomocou tlačidla alebo gesta. Jeho silnou stránkou je, že dokáže obsahovať oveľa väčšie množstvo položiek a je možné k nemu pristupovať z každého miesta webu. Nevýhodou tohto menu je však to, že si užívateľ nemusí uvedomiť jeho prítomnosť, nakoľko vo východnom stave je skryté (Obrázok 6)[23][25].



Obr. 6: Hamburger Menu
[27]

Navigačné centrum - stránka (najčastejšie domovská) aplikácie, ktorá je oddelená od zvyšného obsahu. Z tejto stránky je možné sa navigovať do zvyšku aplikácie, a to prechodom do jej špecifických stránok. Pre navštívenie inej časti aplikácie je nutné, aby sa užívateľ vrátil späť do navigačného centra. Tento prístup je efektívny pre aplikácie, ktoré majú svoj obsah rozdelený do rôznych úloh (Obrázok 7)[23].



Obr. 7: Navigačné centrum
[28]

Výber vhodných navigačných prvkov sa dá vyhodnocovať na základe rôznych faktorov. Pre weby, ktoré nevyžadujú vysoké množstvo možností v navigácii, je vhodný navigačný panel a panel štítkov. Tie sú pre užívateľa ľahko viditeľné a efektívne splňujú svoj účel. Pre weby, ktoré vyžadujú väčšie množstvo možností v navigácii, je doporučené Hamburger Menu alebo navigačné centrum. Voľba medzi týmito dvoma prvkami závisí od povahy stránky. Hamburger Menu je vhodné použiť, keď navigácia obsahuje viacero možností, a pokiaľ ich potrebujeme prístupné kdekoľvek v aplikácii. Je ale nutné počítať s tým, že nie vždy si užívateľ uvedomí jeho prítomnosť. Využitie navigačného centra sa doporučuje pre weby, ktoré potrebujú v navigácii množstvo možností, ktoré vyžadujú svoj vlastný priestor a sú tzv. task-based ⁵. Nevýhodou je v jeho prípade len neustála nutnosť sa doň vracieť, pokiaľ chceme navštíviť inú časť webu[23].

V našej aplikácii sú využité dva druhy navigácii. Hamburger Menu, ktoré je využívané na navigáciu v rámci užívateľských projektov, harmonogramov a ich miest. Zvolené bolo preto, nakoľko množstvo projektov a ich položiek môže byť obsiahle a mali by byť prístupné v rámci každej časti webu. Druhým navigačným prvkom je navigačný panel. Pomocou neho je užívateľ schopný jednoducho zistiť prítomnosť a pristupovať k Hamburger Menu. Umožňuje tiež navigáciu na domovskú stránku webu, ktorá obsahuje interaktívnu mapu.

⁵jedná sa o weby, ktoré svoj účel dosahujú pomocou množiny oddelených úloh

4 Návrh a implementácia aplikácie

Táto kapitola sa zameriava na konkrétne detaily návrhu a implementácie aplikácie. Čitateľovi nieje poskytnutý žiaden úvod k použitiu návrhových a vývojových nástrojov. Pre plné pochopenie tejto kapitoly sa teda očakáva, že čitateľ má aspoň základnú znalosť jazyka TypeScript, HTML, CSS, modelovacieho jazyka UML a objektovo orientovaných princípov.

4.1 Návrh

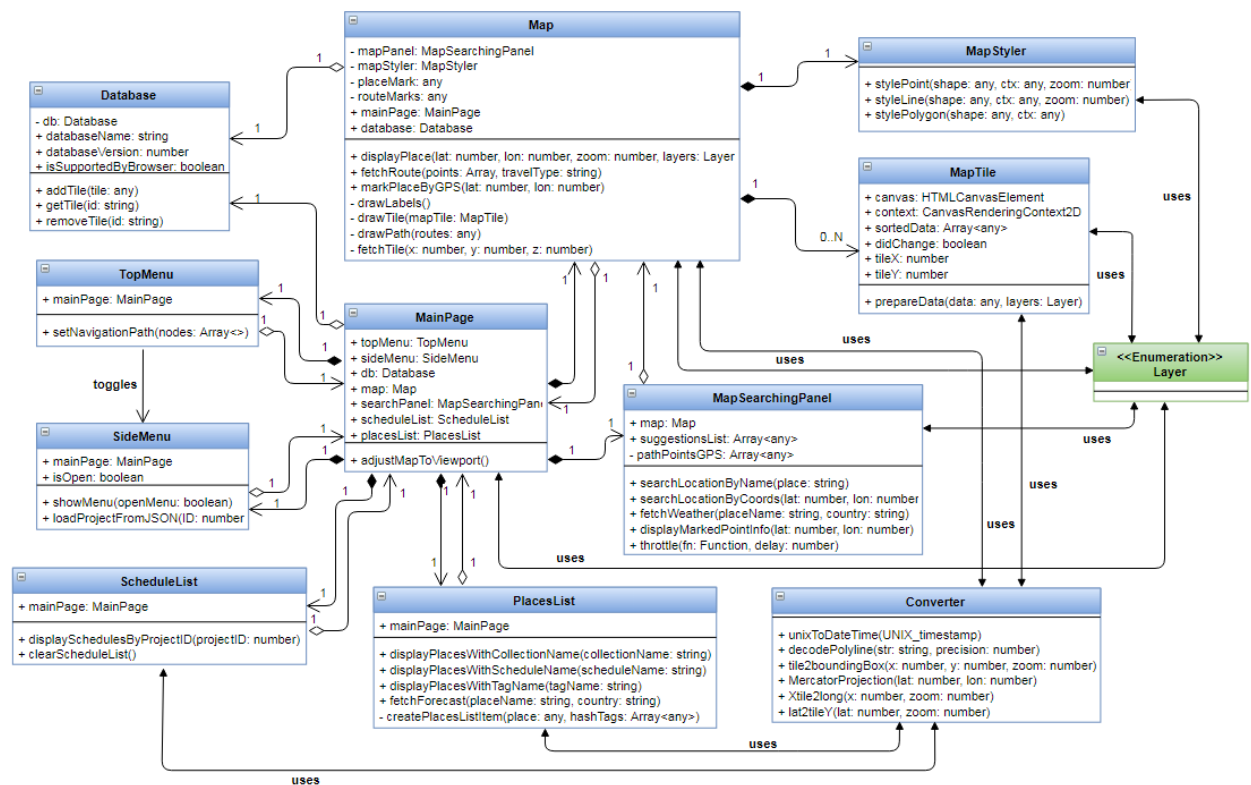
Aplikácia by mala určite obsahovať triedu reprezentujúcu interaktívnu mapu. Táto trieda by sa mala starať o vykresľovanie mapy a operácie spojené s jej funkcionalitou. Pre tento účel je teda vytvorená trieda s názvom `Map`. Tá primárne vykresľuje užívateľom vybrané geodáta a to pomocou elementu `<canvas>`. Jej úlohou je však tieto dáta nielen vykresľovať, ale tiež si ich zaopatriť, a to ako zo serverov tretích strán, tak aj z databázy prehliadača. Okrem toho na mape tiež poskytuje možnosť tvorby tzv. pinpointov ⁶. Medzi týmito pinpointami sa totiž vykresľuje samotná trasa. Ďalšou zodpovednosťou tejto triedy je poskytnúť užívateľovi možnosť interakcie s vykresľovaným obsahom. Preto teda poskytuje metódy, pomocou ktorých je možné vyberať, posúvať a zoomovať zobrazovaný obsah. Prvok mapy kvôli svojmu účelu utvorí približne 75% viditeľnej plochy stránky.

Samotná mapa ale potrebuje prvok, ktorým môže užívateľ vyhľadávať miesta pomocou názvu a zobrazovať ich detailné informácie. Práve k tomuto účelu je vytvorená trieda `MapSearchingPanel`. Obsahuje vstupné pole, ktorým si užívateľ môže vyhľadávať konkrétne miesta, ktoré mapa zobrazí. Okrem toho, pre tieto miesta zaobstaráva a zobrazuje `MapSearchingPanel` dáta o predpovedi počasia. Táto komponenta ma tiež za úlohu poskytovať užívateľovi rozhranie pre vyhľadávanie a zobrazovanie trás. To sa odohráva v rámci viacerých bodov a typov dopravných prostriedkov. Všetky dáta o trasách, miestach a počasí komponenta sťahuje zo serverov tretích strán.

Pomocnou triedou pre interaktívnu mapu je trieda `MapTile`. Jej úlohou je fungovať ako kontajner pre vektorové geodáta mapy. Instancia tejto triedy vždy reprezentuje jeden článok z kolekcie článkov, ktoré vykresľuje instancia triedy `Map`. Vykresľovanie mapy je štylizované pomocnou triedou `MapStyler`, ktorej úlohou je definovať štýl objektov a štítkov kreslených na mapu. Pre povahu vektorových geodát je vytvorený výčtový typ `Layers`. Ten nám slúži ako enumerácia jednotlivých vrstiev, ktoré obsahujú vektorové geodáta. Využitie tohto výčtového typu zjednodušuje selekciu a orientáciu medzi vrstvami, ktoré mapa sťahuje a vykresľuje. Ďalej je vytvorená pomocná trieda `Converter`. Jej účelom je pre celú aplikáciu poskytovať jednoduchý prístup k metódam určeným pre konverziu medzi rôznymi jednotkami. Jedná sa predovšetkým o geografické, teplotné, ale aj o prevody formátov dátumu.

⁶pinpoint - relatívne presne označené miesto (analógia so špendlíkom na nástenke)

Na diagrame tried je možné vidieť úplnú štruktúru objektov aplikácie ⁷. (Obrázok 8).



Obr. 8: Diagram tried predstavujúci domovskú stránku

⁷Obsah tried v diagrame je pre zachovanie jeho čitateľnosti redukovaný na najvýznamnejšie vlastnosti a metódy

Na základe vybraného projektu v **SideMenu** má užívateľ možnosť prezerať si jeho obsah. Základným kameňom štruktúry užívateľských projektov je miesto. Z týchto miest pozostávajú harmonogramy a kolekcie. Kolekcia predstavuje zoznam miest, ktoré majú spoločné znaky (napr. oblasť, v ktorej sa nachádzajú). Užívateľ si tak môže vyhľadať zadané miesta podľa mena kolekcie, do ktorých si ich uložil. Z takýchto miest si môže užívateľ tiež poskladať harmonogram. Ten slúži ako zoznam miest, do ktorých si užívateľ môže naplánovať cestu. Okrem toho tiež obsahuje užívateľské poznámky. Tie môžu napríklad slúžiť pre identifikáciu účelu harmonogramu. Miesta samotné je možné identifikovať nielen podľa názvu kolekcie, v ktorej sa nachádzajú, ale aj podľa štítka, ktorým sú označené.

Očakáva sa však, že detailnejšie dáta užívateľských projektov sú väčšieho objemu, aký je schopný zobraziť prvok **SideMenu**. Pre riešenie tohto problému je teda nutné detailnejšie informácie zobrazovať na väčšej ploche. To je možné buď na špeciálnej stránke, alebo pripojiť obsah k aktuálnej stránke s tým, že pôvodný obsah zakryjeme. Vzhľadom k tomu, že zobrazenie na rovnakej stránke je nielen menej pracné, ale aj menej časovo náročné, bol zvolený takýto prístup. Pre účel zobrazenia takýchto detailnejších informácií projektu boli teda vytvorené triedy **ScheduleList** a **PlacesList**. Ich funkcionality zahŕňa vyhľadávanie, triedenie a zobrazovanie dát projektov na základe zvolených parametrov. Okrem toho má tiež trieda **PlacesList** za úlohu získavať data o počasi pre jednotlivé miesta.

Do sekcie s rozvrhmi projektu je užívateľ presunutý, keď si z prvku **SideMenu** vyberie položku **Schedules**. V rámci zobrazenia tejto sekcie má potom možnosť vidieť zoznam vytvorených harmonogramov. V prípade, že si užívateľ nejaký zvolí, zobrazí mu **MainPage** sekciu, v ktorej si môže miesta uložené v tomto harmonograme prezerať. V tejto sekcii je možné zobrazovať miesta projektu na základe troch rôznych parametrov, a teda na základe výberu harmonogramu alebo pomocou názvu štítku / kolekcie.

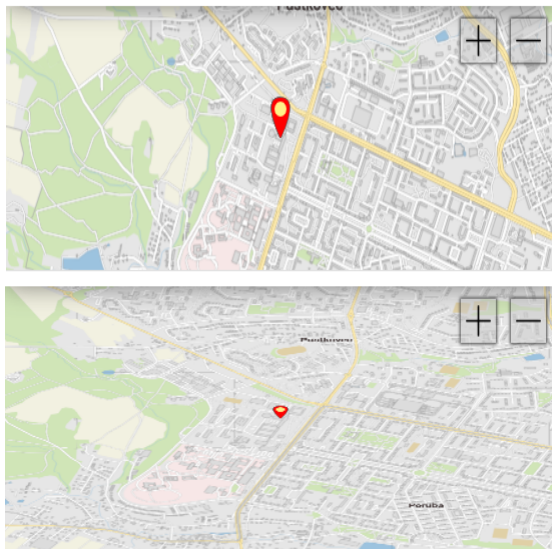
4.2 Implementácia

4.2.1 Trieda **MainPage**

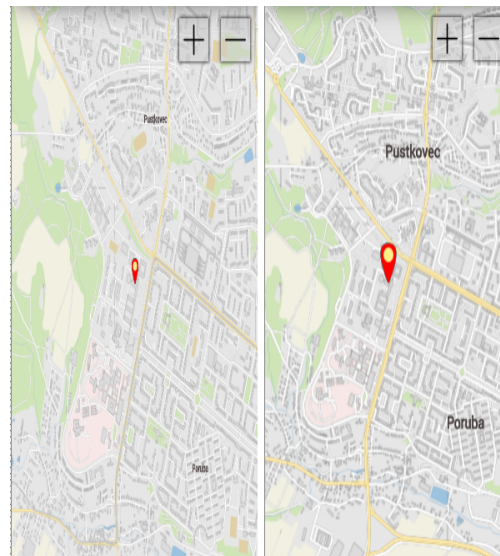
Koreňovým prvkom obsahu domovskej stránky je trieda **MainPage**. Jej hlavnou úlohou je fungovať ako centrum, z ktorého je možné tvoriť zvyšné prvky a ovládať celú stránku. Instancia tejto triedy je vytvorená okamžite po vytvorení Domain Object Modelu (DOM) a načítaní zdrojov. Tvorba obsahu samotnej stránky musí prebiehať v určitom poradí a na jej konci stojí plne použiteľná stránka. V konštrukture sa ako prvé vytvoria všetky objekty užívateľského rozhrania, nasledované inicializovanou instanciou triedy **Database**. Tá je nutná pre správnu funkciu instance triedy **Map**.

V tomto momente je už možné a vhodné mapu vykresliť. Prioritne sa vykresľujú súradnice, ktoré stránka obdržala z inej časti našej aplikácie (napr. z vybraného miesta v projektoch). Inak sa vykreslia východzie GPS koordináty. Pri vykreslenej mape je však problémom zmena dimenzií okna prehliadača, ktorá spôsobuje nesprávne zobrazenie obsahu mapy. Tento problém

je obzvlášť viditeľný pri veľkom rozdiel v pomere výšky a šírky (Obrázky 9a 9b). Je spôsobený východným správaním `<canvas>` elementu, ktorý sa snaží zachovať svoj obsah tým, že ho stlačí do požadovanej veľkosti.



(a) Zobrazenie obsahu na šírku



(b) Zobrazenie obsahu na výšku

Pre zachovanie čítelnosti mapy je však nutné, aby sa zmenili nielen dimenzie elementu `<canvas>`, ale aj dimenzie vykresľovaného obsahu mapy. Toto je implementované pomocou metódy `adjustMapToViewport()` triedy `MainPage`. Tá pri každej zmene veľkosti okna adekvátne upraví obe dimenzie a mapu prekreslí nanovo.

Okrem toho trieda `MainPage` poskytuje implementáciu dotykových gest, a to ako pre aplikáciu, tak pre mapu samotnú. Ako gesto pre otvorenie navigačného `SideMenu` na stránke slúži potiahnutie jedným prstom zľava doprava, gesto v opačnom smere slúži pre jeho zatvorenie. `SideMenu` sa však zatvorí aj v prípade, že užívateľ použije gestá pre pohyb dole či hore. Gestá mapy sú dosť podobné. Pre posun do ľubovoľného smeru na mape stačí jedným prstom potiahnuť do opačnej strany. Zoomovanie na mape je riešené pomocou dvoch prstov. Priblíženie je možné dosiahnuť potiahnutím dvoch prstov od seba, oddialenie zase potiahnutím k sebe.

4.2.1.1 Objekt `Promise`

Prístup do indexovanej databázy nám samozrejme trvá nejaký čas. Preto aby nedochádzalo k blokovaniu celej stránky počas načítavania, je nutné databázu inicializovať asynchrónne. Pre tento účel a tiež celkovo pre každú asynchrónnu operáciu v aplikácii sa využíva objekt `Promise`.

"Promise poskytuje mechanizmus pre plánovanie práce, ktorá má byť vykonaná na hodnote, ktorá prezatiaľ nieje prístupná. Je to abstrakcia pre manažovanie interakcií s asynchrónnymi API.[29]"

Jedná sa o prehľadnejšie a ľahšie udržateľné riešenie callback funkcií. **Promise** obsahuje pre nás dve podstatné funkcie. Prvou je `then(onCompleted)`. Táto funkcia je volaná v prípade, že asynchrónna operácia prebehla úspešne. Druhou funkciou je `catch(onRejected)`, a tá je volaná v prípade, že asynchrónna operácia zlyhala. Tieto operácie je možné reťaziť, a tým zachovať prehľadnosť i pri väčších množstvách asynchrónnych operácií[29].

4.2.2 Trieda **MapTile**

Instancie tejto pomocnej triedy nám slúžia pre prehľadné uchovanie jednotlivých dlaždíc geodát. Okrem dát samotných obsahuje **MapTile** informácie opisné a tiež čiastočne spojené s vykresľovaním. Jedná sa napr. o pozíciu v dlaždicovom systéme geodát, pozíciu na mape, GPS koordináty, úroveň zoomu, dimenzie dlaždice, koeficient pre správne škálovanie vykreslenia v rámci týchto dimenzií a mnoho podobných.

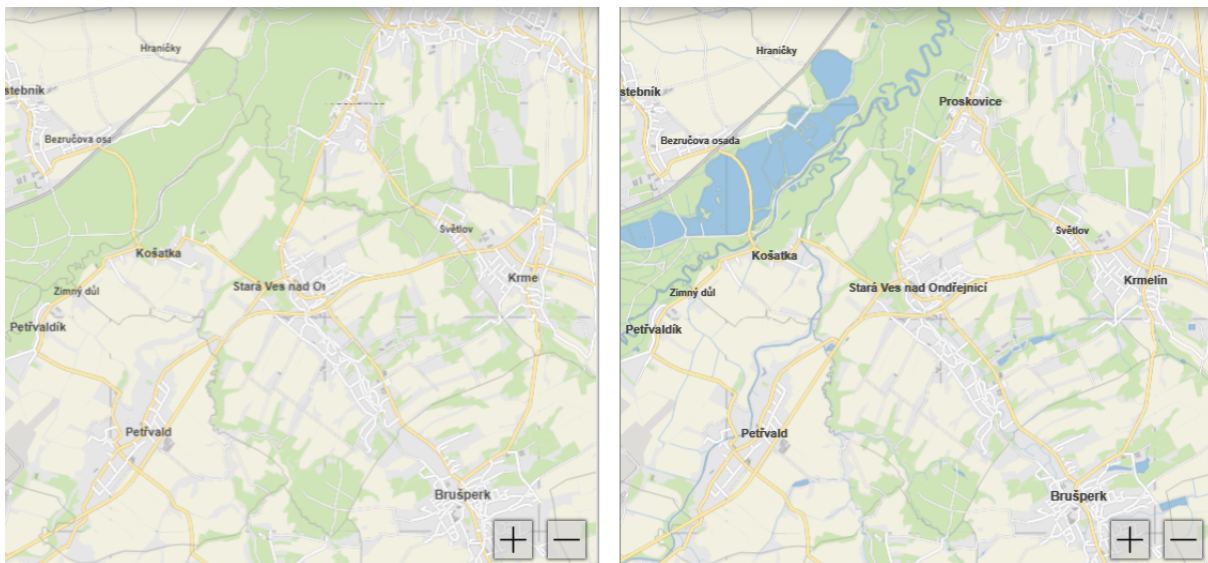
Medzi hlavné objekty v objekte **MapTile** patrí canvas typu **HTMLCanvasElement** a context typu **CanvasRenderingContext2D**. Tieto prvky poskytujú možnosť cachovania vykreslených dát. Pri počiatočnom vykreslení sa totiž do **HTMLCanvasElement** pomocou **CanvasRenderingContext2D** vykresľuje geometria geodát. Objekt **HTMLCanvasElement** je do seba schopný toto vykreslenie uložiť a udržiavať vo forme obrázku. Nakoľko **CanvasRenderingContext2D** obsahuje aj metódy pre kreslenie obrázkov, je potom vykreslenie celej dlaždice otázkou vykreslenia jediného obrázku. Využitie tejto mechaniky je pre nás obzvlášť prínosné pri posune mapy. Každý posun gestom totiž vyžaduje zmazanie a vykreslenie plochy mapy nanovo (cca 5-10 krát, počet závisí na rýchlosti gesta).

Táto trieda má tiež na starosti podľa kľúča usporiadať kolekciu geodát, ktorú v sebe uchováva. Je totiž nutné, aby objekty boli vykresľované v správnom poradí. Poskytované geodáta sú prvotne neusporiadané, čo by mohlo znamenať, že pri ich vykreslení sa môžu objekty vrstvy prekrývať nesprávne (Obrázok 9).

4.2.2.1 Vrstvy mapy

Vykresľovanie mapy sa odohráva na dvoch vrstvách. Prvá (spodná) vrstva má za úlohu vykresľovať objekty mapy ako sú uzemia, budovy a cesty. Druhá vrstva má za úlohu vykresľovať názvy budov, miest, obcí, štátov a vodných zdrojov. Každá z týchto vrstiev je vykresľovaná na svoj vlastný `<canvas>` element.

Dôvodom k použitiu viacerých vrstiev je prekrývanie jednotlivých názvov miest. Nakoľko spodná vrstva vykresľuje dáta po častiach, ktoré sú dodávané asynchrónne, nie je možné zaručiť predpovedateľné poradie. Pri názvoch, ktoré sú umiestnené na hrane dlaždice tak môže dôjsť k prekrytiu susednou dlaždicou, ktorá sa vykreslila neskôr. Toto je demonštrované na obrázku č.9 . Preto bola implementovaná druhá (vrchná) vrstva, ktorá po prvej vrstve vždy vykresľuje názvy do jedinej dlaždice o veľkosti celej mapy.



Obr. 9: Rozdiel medzi poradím vrstiev a problém vykreslenia názvov

4.2.3 Trieda Map

Jedná sa zrejme o najdôležitejšiu a implementačne najzložitejšiu triedu aplikácie. Medzi najhlavnejšie prvky tejto triedy patrí kolekcia s instanciami `MapTile`. Z tej prebieha vykresľovanie a manipulácia s geodátami. Nemenej dôležitý je objekt `Database`, ktorý nám umožňuje k objektom `MapTile` pristupovať alebo ich ukladať. Trieda tiež obsahuje kolekciu, ktorá udržiava informácie o užívateľom vytvorených pinpointoch. Pre vykresľovanie existujú v triede objekty `HTMLCanvasElement` a `CanvasRenderingContext2D`.

4.2.3.1 Vykresľovanie mapy

Vektorové geodáta sú obsiahleho charakteru a ich vykresľovanie je netriviálnym procesom. Preto je ich kreslenie nutné rozložiť na menšie časti. Základným predpokladom pre tento účel je rozdelenie dát do blokov, dlaždíc, čo je realizované pomocou triedy `MapTile`. Takéto bloky je však stále nemožné vykresliť naraz. Preto je potrebné pristupovať k nim na elementárnej úrovni, teda na úrovni geometrických útvarov.

Na celý proces vykresľovania sa dá pozeráť ako na strom, pričom koreňom tohto stromu je v našom prípade metóda:

```
displayPlace(latitude: number, longitude: number, zoom: number, layers: Layer)
```

Tá pre nás predstavuje rozhranie pre vykreslenie určitej oblasti pomocou predaných súradníc, vrstiev a úrovne zoomu. Konkrétnou funkciou tejto metódy je však pripraviť mriežku objektov `MapTile` pre vykreslenie zadaných hodnôt, ale aj mapu pre dynamické načítavanie ďalších dlaždíc. Spôsob, ktorým táto metóda realizuje funguje v skratke takto:

1. Premaže sa plocha mapy a kolekcia obsahujúca geodáta z predchádzajúceho vykreslenia.

2. Aktualizujú sa zobrazované hodnoty (úroveň zoomu, GPS súradnice, vrstvy atď.)
3. Do stredu mapy sa umiestni `MapTile` s predanými koordinátami a vypočíta sa počet dlaždíc pre vyplnenie zvyšnej plochy.
4. Všetky dlaždice mapy sa vytvoria a uložia do kolekcie geodát ako placeholder ⁸.
5. Metóda `fillPlaceholder` sa zavolá na všetky `MapTile` okrem vonkajšej vrstvy.

Úlohou metódy `fillPlaceholder(tileToFill: MapTile)` je rozhodnúť, z ktorého zdroja predaná `MapTile` získa svoje geodáta a následne ich vykresliť. Preferovaným zdrojom je vždy `Database`. Pokiaľ však databáza požadované geodáta neobsahuje, nechá ich metóda stiahnuť asynchrónne zo servera.

Samotné vykreslenie dlaždice je realizované pomocou metódy `drawTile(mapTile: MapTile)`. Placeholder, teda `MapTile` bez geodát sa nikdy nevykresľuje. Ak `MapTile` obsahuje geodáta, je ešte nutné overiť či boli pozmenené. Pokiaľ sú nezmenené, tak sa využíva už spomínaného cachovaného vykreslenia (Kapitola 4.2.2) pomocou `HTMLCanvasElement`-u tejto dlaždice. V opačnom prípade sa začne vykresľovanie kolekcie geodát.

Pomocou metód pre vykreslenie špecifických geometrických útvarov sa následne vykreslí celá kolekcia geodát. Každý útvar sa zároveň štýluje adekvátne k objektu, ktorý má predstavovať. Pre projekciu geodát sa pritom používa Merkátorova projekcia.

4.2.3.2 Ovládanie mapy

Užívateľ však bude tiež chcieť vykreslenú mapu posúvať alebo zoomovať. Ako už bolo zmienené v návrhu, táto interakcia je realizovaná pomocou gest v triede `MainPage`. Riešenie zoomovania mapy je pre nás vďaka parametrom, ktoré prijíma metóda `displayPlace` triedy `Map`, relatívne triviálnou záležitosťou. Pri vykonaní gesta pre zoom sa pomocou metódy `getCoordinatesAtCenter()` triedy `Map` zistia GPS súradnice v strede mapy. Tie sú následne vložené do už spomenutej metódy `displayPlace`, pričom sa zmení úroveň zoomu vzhľadom k vykonanému gestu.

Posun mapy do strán už je o niečo komplikovanejší. Pre ten nám slúži v triede `Map` metóda `shiftMap(shiftX: number, shiftY: number)`. Telo metódy je možné vidieť na výpise 1.

```

1  public shiftMap(shiftX: number, shiftY: number) {
2      for (var i = 0; i < this.mapData.length; i++) {
3          this.mapData[i].positionX += shiftX;
4          this.mapData[i].positionY += shiftY;
5          this.mapData[i].isRendered = false;
6      }
7      this.clearCanvas();
8      for (var i = 0; i < this.mapData.length; i++) {
9          this.drawTile(this.mapData[i]);
10     }

```

⁸jedná sa o objekt, ktorého jediný účel je držať miesto pre svojho nástupcu

```
11  this.drawPath(this.currentRoute);
12  this.redrawAllMarks();
13  this.updateMap();
14  }
```

Výpis 1: Implementácia posunu mapy do strán

Táto metóda je volaná pri každom posuve mapy. Parametre predstavujú rozdiel v súradniciach, o ktorý sa má mapa posunúť vzhľadom k poslednej polohe. Tento rozdiel je pripočítaný ku každej dlaždici v kolekcii `mapData` triedy `Map`, čím sa dosiahne posun celej mapy. Potom je už len nutné vyčistiť predošlý obsah a vykresliť upravené geodáta nanovo. Na vykreslený obsah sa potom kreslia cesty a pinpointy, ktoré sú objasnené v kapitole 4.2.3.4 .

Ako posledný krok sa obsah kolekcie `mapData` aktualizuje pomocou metódy `updateMap()`.

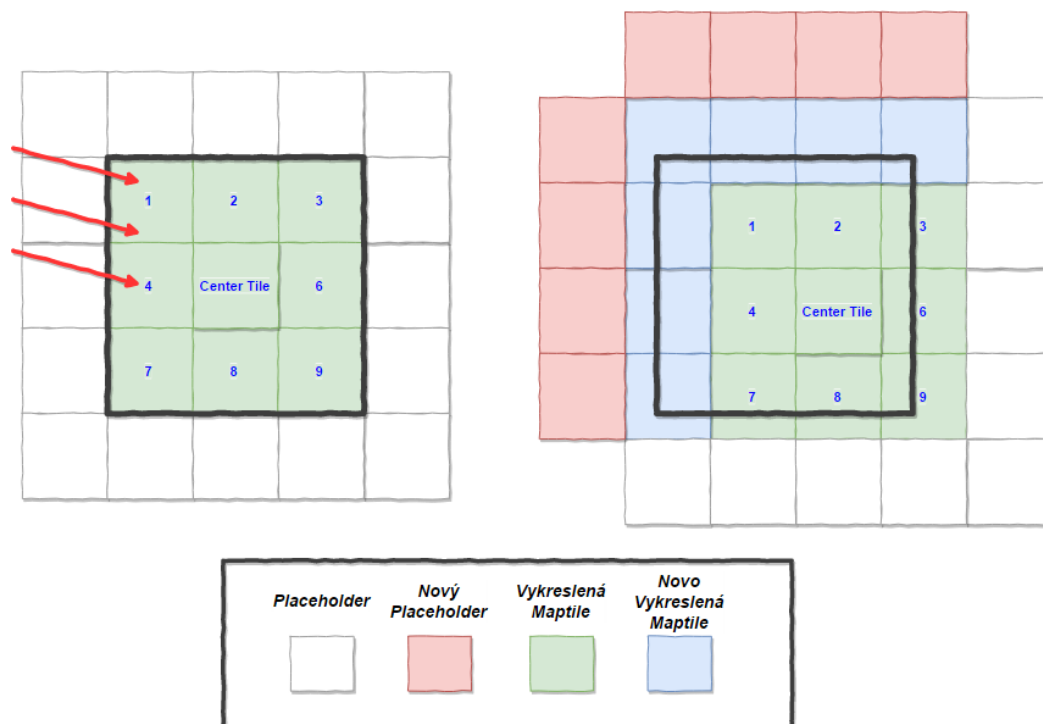
4.2.3.3 Dynamické načítanie obsahu

Zmysel placeholder objektov je práve v realizácii dynamického načítavania obsahu. Toto načítavanie sa odohráva pri každom posuve mapy pomocou metódy `updateMap()`. Jej úlohou je totiž vyhľadať všetky placeholder objekty, ktoré sa vďaka posunu mapy vyskytujú na viditeľnej ploche. Pre každý takýto placeholder však najprv zistí, či na všetkých jej stranách (okrem diagonálnych) susedí s nejakou `MapTile`. Pokiaľ na nejakej strane nemá suseda, vytvorí a umiestni sa na túto stranu nový placeholder. Keď má náš placeholder v každom smere suseda, nechá sa vykresliť pomocou už spomínanej metódy `fillPlaceholder()`. Týmto spôsobom sa na viditeľnej ploche vždy dokreslia potrebné geodáta. Vonkajšiu vrstvu mriežky mapy pritom tvoria stále placeholder objekty. A teda v prípade ďalšieho posunu celý tento proces prebieha znova. Na užívateľa to tak môže pôsobiť, že obsah mapy je kontinuálny a on má možnosť vidieť len časť z nej. Tento proces je ilustrovaný na obrázku 10. Naľavo sa nachádza mapa pred posunom, na pravej strane po posune.

4.2.3.4 Vykreslenie pinpointov a trasy

Viditeľnosť týchto objektov má v rámci spodnej vrstvy mapy najvyššiu prioritu. Preto ich vykresľovanie prebieha vždy až po vykreslení geodát mapy. Pre vytvorenie pinpointu užívateľom si vystačíme s koordinátmi dotyku na ose mapy. My však ale potrebujeme pinpoint uložiť a pri prípadnom posune mapy ho vykresliť na správnom mieste. Pre tento účel sa nemožno orientovať podľa súradníc mapy, ale podľa GPS súradníc.

Pre ich získanie je potrebné nájsť `MapTile`, na ktorej pinpoint prvotne vznikol. Pomocou škálovacích koeficientov, ktorá táto `MapTile` obsahuje, sme schopný reverzne dopočítať GPS súradnice. Pinpoint spárovaný s GPS koordinátmi môžeme pri posune mapy vykresliť podobným spôsobom, ako všetky ostatné geometrické útvary. Stačí iba zistiť, v ktorej `MapTile` sa nachádzajú súradnice vykresľovaného pinpointu a použiť jej škálovacie koeficienty. Toto sa deje



Obr. 10: Dynamické načítanie obsahu pri posune mapy

napríklad v metóde `markCollectionRedraw()`, ktorá týmto spôsobom prekresľuje všetky pin-pointy na mape.

Vykresliť trasu medzi pinpointami, ktoré majú GPS súradnice je pomerne jednoduché. Pre API tretej strany, ktoré nám má trasu vyhľadať stačí poskytnúť kolekciu GPS súradníc jednotlivých uzlov. Server nám potom navráti objekt, ktorý túto trasu obsahuje. Nakoľko je trasa zakódovaná vo formáte `polyline` je nutné ju najprv dekodovať. Tento formát výrazne redukuje celkovú veľkosť objektu trasy, čo je obzvlášť výhodné pri trasách s väčším počtom uzlov [30]. Následne už len stačí overiť, či body trasy ešte stále existujú, a ak áno, tak trasu vykresliť ako `MultiLineString`.

4.2.4 Trieda MapSearchingPanel

Táto trieda obsahuje doplnkovú funkcionálnosť pre triedu `Map`. Tá spočíva v poskytovaní užívateľského rozhrania pre štyri druhy funkcionálnosti:

- Vyhľadávanie miest na mape pomocou textu
- Zobrazenie detailov miesta
- Niekoľkodňová predpoveď počasia pre vybrané miesto
- Vyhľadávanie trasy a informácie o nej

Pre účel vyhľadávania je do `MapSearchingPanel` umiestnené vstupné pole. To pomocou asynchrónnej metódy navrhuje užívateľovi najbližšie výsledky odvodené zo zadaného vstupu. Tieto návrhy sú do našej aplikácie sťahované z aplikačných rozhraní tretích strán.

Je bežnou situáciou, že užívateľ ktorý pozná názov hľadaného miesta, sa snaží ho zadať čo najrýchlejšie. Vďaka tomu sa návrhy súvisiace so začiatkom vyhľadávanej frázy stávajú veľmi rýchlo irelevantné. Tak sa stáva pomerne neefektívnou aj sieťová komunikácia za týmto účelom. Pre riešenie tohto problému je v triede `MapSearchingPanel` implementovaná metóda `throttle(fn: Function, delay: number)`. Jej použitie je možné vidieť na nasledujúcom útržku kódu 2.

```
1 .on("keyup", this.throttle(() => {  
2   var text = $("#placeInput").val();  
3   var coords = this.map.getCoordinatesAtCenter();  
4   this.fetchSuggestions(text, coords.latitude, coords.longitude)  
5     .then((suggestions) => {  
6       this.setSuggestions(suggestions);  
7     })  
8   }, 250));
```

Výpis 2: Obmedzenie počtu vyžiadaných sugescií na základe zadaného vstupu

Metóda `throttle` pridáva pre event handler, ktorý spracúva udalosti spojené so vstupom, ďalšiu vrstvu funkcionality. Tá spočíva v obmedzení počtu volaní za určitý interval. V našom prípade je to na jedno volanie za 250ms. Vďaka tomu sa pri rýchlom užívateľskom vstupe vyžiada od servera len zlomok požiadavkov. Zredukuje sa teda počet nepotrebných komunikácií, no stále budú užívateľovi poskytnuté relevantné návrhy.

Užívateľ po vyhľadaní miesta môže vidieť jeho názov, koordináty a predpoveď počasia na 4 dni. Pre vyhľadávanie trasy má užívateľ viacero možností. Môže vyhľadávať trasy medzi viacerými bodmi, pričom tieto body si môže vyhľadať pomocou ich názvov. Toto hľadanie je tiež podporené funkciou autocomplete. Užívateľ si pritom môže vybrať či si praje optimalizovanú trasu pre dopravu autom, bicyklom alebo pešo. Po vyhľadaní sú užívateľovi poskytnuté aj informácie o dĺžke a časovej náročnosti cesty.

V rámci autocomplete služby sa použitím koordinátov zobrazenej lokácie snažíme čo najpresnejšie odhadnúť hľadajú lokáciu podľa názvu. Bohužiaľ, v niektorých prípadoch sa stáva, že náš datový zdroj nevie správne vyhodnotiť hľadané miesto na základe textu. Problémom pritom zvyčajne býva viacero miest z rovnakým názvom.

4.2.4.1 Užívateľová geolokácia

V triede `MapSearchingPanel` je implementovaná tiež možnosť vyhľadania trasy na základe užívateľovej geolokácie. Táto služba však v niektorých populárnych moderných prehliadačoch⁹ vyžaduje aby stránka, ktorá ju využíva, komunikovala skrz zabezpečený protokol HTTPS. V našom prípade datový zdroj poskytujúci predpoveď počasia však HTTPS komunikáciu nepodporuje. Bolo preto potrebné toto obmedzenie obísť pomocou využitia proxy skriptu. Toho úlohou je sťahovať na server dáta pomocou HTTP, ktoré poskytuje klientskej aplikácii cez HTTPS. V rámci klienta sa komunikácia stala zabezpečenou, čo užívateľovi umožňuje vyhľadávanie na základe svojej geolokácie.

4.2.5 Datové zdroje aplikácie

Využitie otvorených zdrojov tretích strán nám poskytuje jednoduchý a spoľahlivý prístup ku geodátam a dátam s nimi spojenými. Všetky použité dátové zdroje využívajú pre komunikáciu REST-ful API. Táto technológia využíva oproti ostatným, robustnejším komunikačným modelom ako napr. SOAP, oveľa menšiu šírku pásma. Pre našu aplikáciu je tak vhodnejšou [31].

Pre prístup k dátam o miestach a autocomplete návrhom je využitá služba `MapzenSearch` [32]. Z tejto REST-ful služby sú v našej aplikácii využité tieto API:

Autocomplete - poskytuje návrhy miest, umožňuje tak užívateľovi rýchlejší výber miesta [32].

Forward geocoding - umožňuje preklad názvov miest a ulíc na GPS koordináty [32].

Reverse geocoding - ide o preklad GPS koordinátov na názvy miest či ulíc [32].

Optimalizované trasy pre našu aplikáciu poskytuje služba `Mapzen Mobility`. Trasy je možno prepočítavať pre rôzne spôsoby dopravy. Jedná sa napríklad o auto, bicykel, pešiu chôdzu, verejnú dopravu a kombinácie uvedených. Bezplatná verzia tejto služby nám však dokáže poskytnúť trasu medzi bodmi, ktoré sú vzdialené nanajvýš 200km [30].

K vektorovým geodátam sa pristupuje pomocou služby `Vector Tile Service`. Bezplatná verzia tejto služby síce ovplyvuje určitými limitmi, no tie sú pre účel našej aplikácie zanedbateľné. Geodáta poskytuje vo formáte `GeoJSON`, `TopoJSON` a `Mapbox-format binary tiles` [33]. V našej aplikácii sa používa formát `GeoJSON`.

Dáta o počasí sú pre aplikáciu poskytované pomocou služby `OpenWeatherMap`. Hoci táto služba poskytuje širokú škálu dát, pre účely tejto aplikácie si vystačíme s:

⁹napr. Chrome, Chrome for Android, Opera, iOS Safari, Android Browser

Current weather data - poskytuje dáta o aktuálnom počasi v zadanej lokácii.

5-day weather forecast - umožňuje sťahovať predpoveď počasia pre zadané miesto na 5 dní dopredu.

4.2.6 Triedy `PlacesList` a `ScheduleList`

Pomocná trieda `PlacesList` má za úlohu zobrazovať zoznamy miest. Tie dokáže zobrazíť na základe názvu kolekcie, štítku alebo názvu rozvrhu, v ktorom sú umiestnené. Do každého takéhoto miesta v zozname sú vložené informácie o názve, polohe, popise a odhadovanom čase na návštevu. Pre jednotlivé miesta je tiež poskytovaná predpoveď počasia z už uvedených zdrojov tretích strán. V rámci miesta sú užívateľovi poskytované aj fotografie danej oblasti. Pre zobrazenie kolekcie takýchto fotografií sa používa prvok `Carousel`¹⁰. V našom prípade sa konkrétne jedná o `Carousel` z knižnice `Slick`. [34] Takýto prvok periodicky prechádza a zobrazuje jednotlivé fotografie s poskytnutej kolekcie, čím značne šetrí miesto na stránke.

Miesta v zozname miest ponúkajú tiež možnosť zobrazenia ich lokácie na mape po kliknutí na príslušné tlačidlo. Užívateľ si tak môže rýchlo a jednoducho prezrieť miesto na mape. Pre rýchly návrat do prezeranej kolekcie miest je následne na mapu umiestnené špeciálne tlačidlo, ktoré ho tam presunie.

Účelom triedy `ScheduleList` je zase zobrazíť kolekciu rozvrhov pre výlety. Každý zobrazený rozvrh poskytuje užívateľovi informácie o výlete ako napr. čas započatia i ukončenia výletu, poznámky a počet miest k navštíveniu. Kliknutím na špecifické tlačidlo môže následne užívateľ prezeráť v triede `PlacesList` jednotlivé miesta rozvrhu.

4.2.6.1 `WebStorage`

Naša aplikácia sa skladá z jedinej stránky, v ktorej sa prepína medzi viacerými sekciami. Nakoľko je tu možnosť, že užívateľ danú stránku nechá načítať nanovo, je nutné si pamätať, ktorá sekcia má byť po obnovení zobrazená. Pre tento účel je využitá technológia `WebStorage`. Tá poskytuje dva spôsoby uchovávania údajov stránky [35]:

`localStorage` - poskytuje perzistentné úložisko. K dátam možno pristupovať aj po zatvorení a opätovnom otvorení prehliadača.

`sessionStorage` - funguje ako dočasné úložisko. Dáta sú zmazané už po zatvorení všetkých kariet navštívenej domény.

Informácia o zobrazenej sekcii sa ukladá do `sessionStorage` pod určitým kľúčom. Po opätovnom načítaní karty našej stránky sa na základe hodnoty tohto kľúča zobrazí posledná zobrazená sekcia. V prípade, že však užívateľ zatvorí prehliadač a navštívi stránku znova, zobrazí sa

¹⁰ alias Slideshow alebo Slider

mu východzia sekcia, čo je aj zámerom autora. Rovnaký princíp je použitý aj pre zapamätovanie si poslednej zobrazenej lokácie na mape.

Posledne zvolený projekt je naproti tomu uložený do `localStorage`. Je totiž žiadúce, aby si užívateľ nemusel nastavovať tento projekt pri každom spustení prehliadača.

5 User Experience testovanie

Završením vývojovej fázy sa aplikácia stala vhodnou pre testovanie. Zámerom testovania bolo čo najpresnejšie zistiť použiteľnosť aplikácie z užívateľskej perspektívy. Pre tento účel bolo testovanie vykonané menšou skupinou neprofesionálnych testerov. Tí mali najprv za úlohu sa samostatne zoznámiť a orientovať v prostredí našej aplikácie. Neskôr táto skupina tiež obdržala radu úloh, ktorých účelom bolo otestovať, či sú užívatelia schopní svojpomocne využívať možnosti aplikácie.

5.1 Testované zariadenia

Aplikácia bola testovaná na viacerých zariadeniach i prehliadačoch. Medzi testované zariadenia patria:

- Xiaomi Redmi 4A s operačným systémom Android verzie 6.0.1
- Iphone 5SE s operačným systémom iOS
- Sony Xperia Z5 s operačným systémom Android verzie 7.0

Funkčnosť aplikácie na týchto zariadeniach bola testovaná v prehliadačoch Google Chrome, Chrome For Android a Safari. Aplikácia je implementovaná pomocou responzívneho designu. Nemala by mať teda problém korektne zobrazíť svoj obsah aj na tabletoch či desktopových platformách.

Tabuľka 1: Tabuľka podporovaných technológií v jednotlivých verziách prehliadačov [36]

	<i>HTML5 Datalist</i>	<i>IndexedDB</i>	<i>WebStorage</i>	<i>Canvas</i>
<i>IE</i>	11	11	11	11
<i>Edge</i>	15	15	15	15
<i>Firefox</i>	53	53	53	53
<i>Chrome</i>	58	58	58	58
<i>Safari</i>	10.1	10.1	10.1	10.1
<i>Opera</i>	44	44	44	44
<i>Android Browser</i>	56	56	56	56
<i>Android Chrome</i>	57	57	57	57
<i>Plná podpora</i>	9.52 %	86.18 %	95.22 %	94.88 %
<i>Čiastočná podpora</i>	84.44 %	7.43 %	0.04 %	3.08 %
<i>Podpora celkovo</i>	93.96 %	93.61 %	95.26 %	97.96 %



Plná podpora



Čiastočná podpora



Žiadna podpora

5.2 Podpora v prehliadačoch

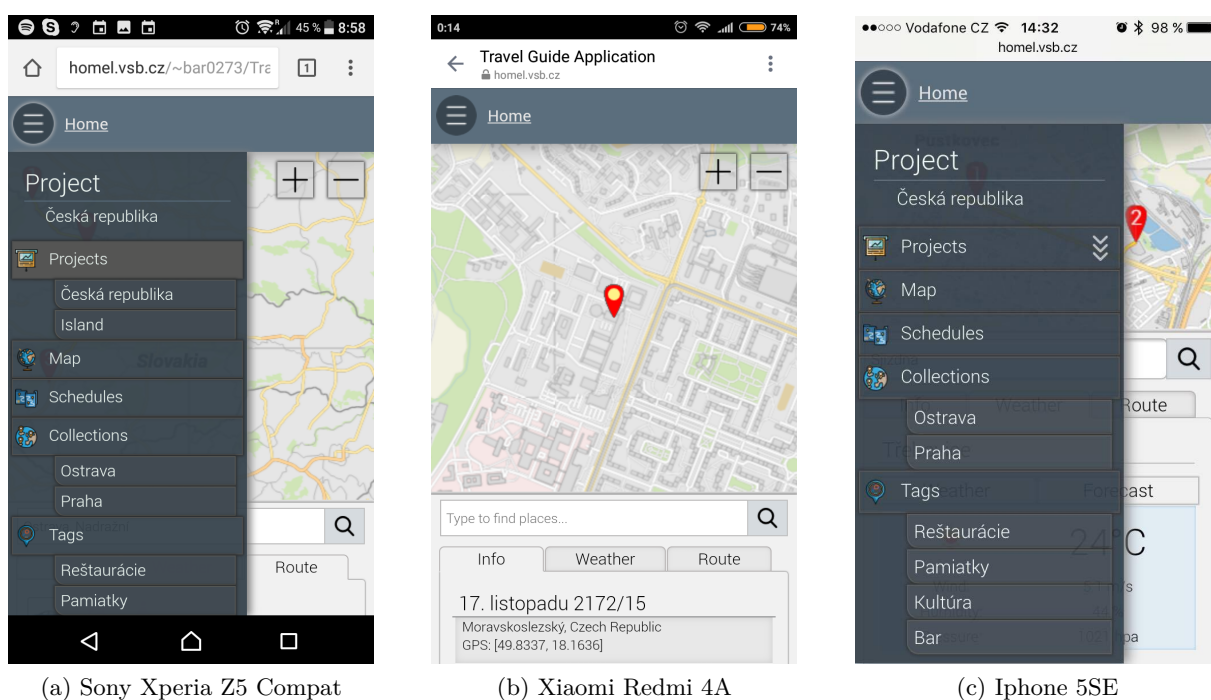
Funkcionalita aplikácie využíva viacero pomerne nových technológií. Nieje teda možné zaručiť, že aplikácia pobeží správne na každom prehliadači. Drvivá väčšina však nami použité technológie podporuje. Bližšie informácie o tejto podpore môžeme vidieť na tabuľke 1.

5.3 User experience

Väčšina užívateľov je schopná sa na prvý pohľad rýchlo zorientovať v rozhraní našej aplikácie. Tomu z veľkej časti napomáha fakt, že sme pri návrhu minimalizovali počet prvkov v rozhraní aplikácie.

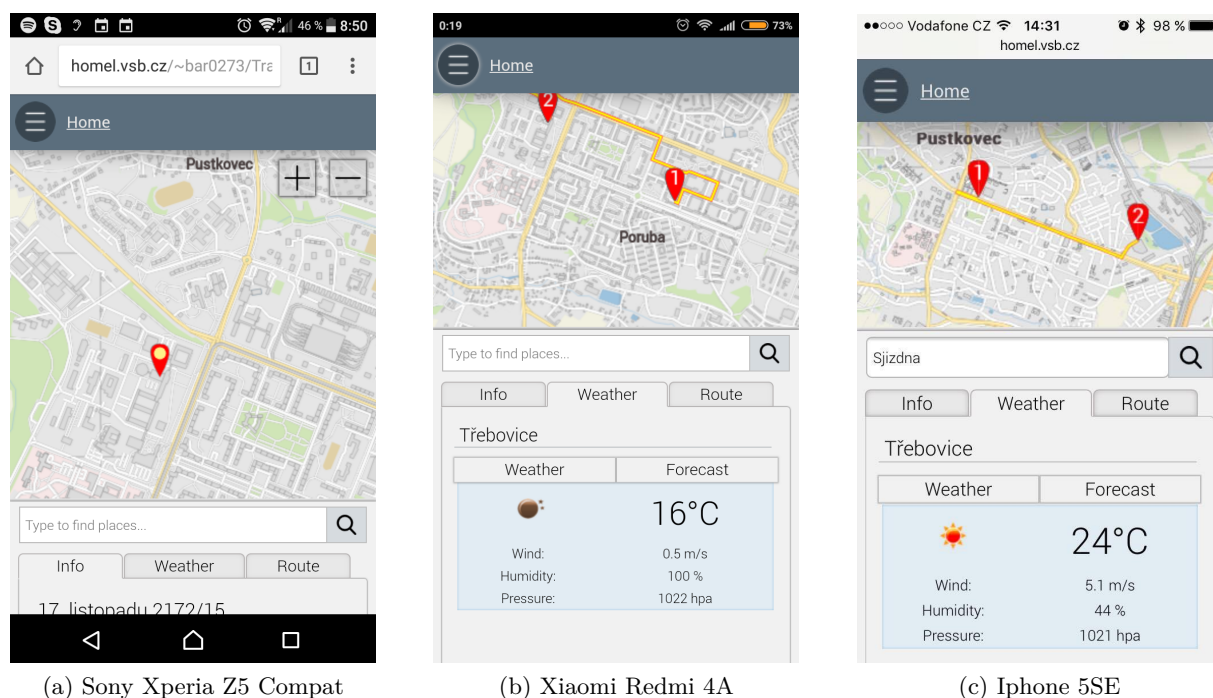
Užívatelia nemajú problém nájsť bočné menu, avšak väčšina si spočiatku neuvedomí, že sa k nemu dá pristupovať aj pomocou gest. Čo sa týka štruktúry projektov, väčšia časť skupiny nemala problém nájsť projekt, zvoliť si ho a pokračovať v skúmaní jeho štruktúry a obsahu. Pre zvyšných užívateľov by bolo však vhodné implementovať napr. krátky tutoriál. Ten by mal za úlohu objasniť používanie projektov, ich význam a štruktúru. Navigácia implementovaná vo vrchnom menu bola pre užívateľov intuitívna, nakoľko sa jedná o štandardný prvok mobilných aplikácií.

Toto testovanie vykonané na rôznych zariadeniach je možné vidieť na nasledujúcich obrázkoch 11a 11b 11c.



Obr. 11: Ilustrácia použitia rozhrania domovskej stránky užívateľom

Následne mali užívatelia za úlohu vyhľadať miesto na mape. Všetci pochopili, že k tomu budú potrebovať vyhľadávacie pole, ktoré je umiestnené vedľa mapy. Vyhľadávané boli miesta i adresy. Užívatelia mohli svoje vyhľadanie potvrdiť buď pomocou virtuálnej klávesnice mobilu alebo vyhľadávacím tlačidlom vedľa vstupného poľa. Pre užívateľov nebolo problémom nájsť informácie o predpovedi počasia. Tento test je možno vidieť na obrázkoch 12a 12b 12c.



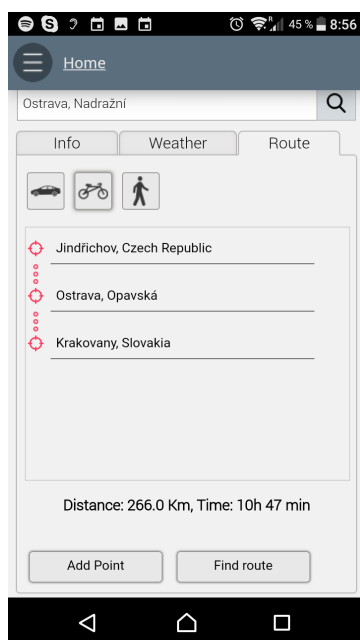
Obr. 12: Ilustrácia vyhľadávania miest na mape užívateľom

Potom mali užívateľa za úlohu pracovať s mapou. Posun mapy gestami prebiehal bezproblémovo. Zoomovanie pomocou gest však pri menších rozlíšeníach bolo obtiažne a pomerne frustrujúce. Tento problém bol odstránený pridaním tlačidiel pre zoomovanie na mapu.

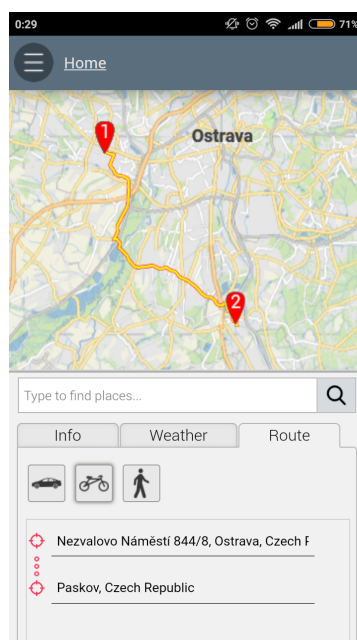
Užívatelia mali však problém jednoznačne indentifikovať niektoré objekty na mape, nakoľko názvy niektorých oblasti sa prekrývajú. Rozmiestnenie názvov by mohlo byť riešené pomocou implementácie jedného z algoritmov určených pre odstránenie kolízií štítkov. Taktiež by bolo vhodné poskytnúť užívateľovi legendu mapy. Vďaka nej by sa predchádzalo nejasnostiam pri identifikácii objektov mapy.

Pre užívateľov bolo na niektorých miestach obtiažné vyhľadanie trasy. API, ktoré poskytuje preklad názvov miest na GPS súradnice totiž pri viacerých totožných názvoch nedokáže vybrať správny. Riešením tohto problému by mohlo byť použitie iného API. Pri zariadeniach s menším rozlíšením by tiež mohlo byť vhodné pri stlačení tlačidla pre vyhľadanie trasy presunúť užívateľa na viditeľnú plochu mapy. Takouto spätnou väzbou by užívateľ mohol okamžite vedieť, že jeho akcia mala nejaký účinok.

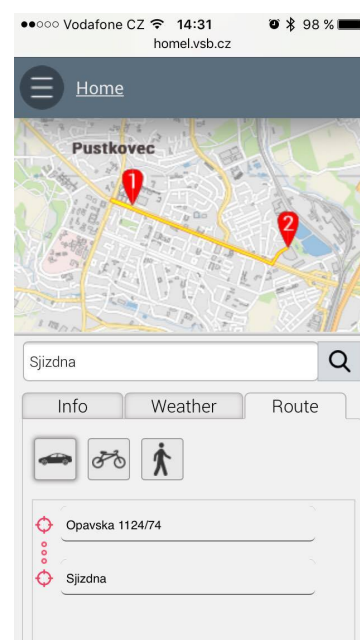
Vyhľadávanie trasy je demonštrované na obrázkoch 13a 13b 13c.



(a) Sony Xperia Z5 Compat



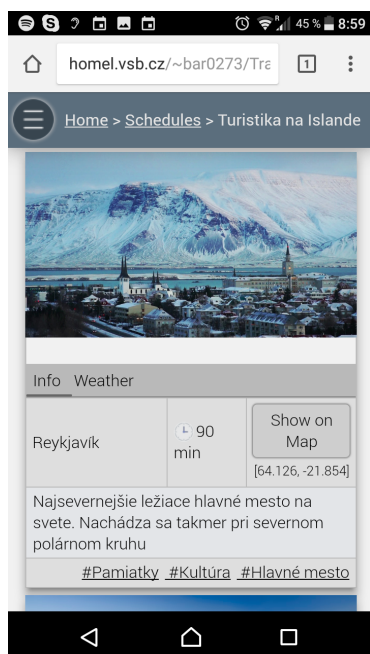
(b) Xiaomi Redmi 4A



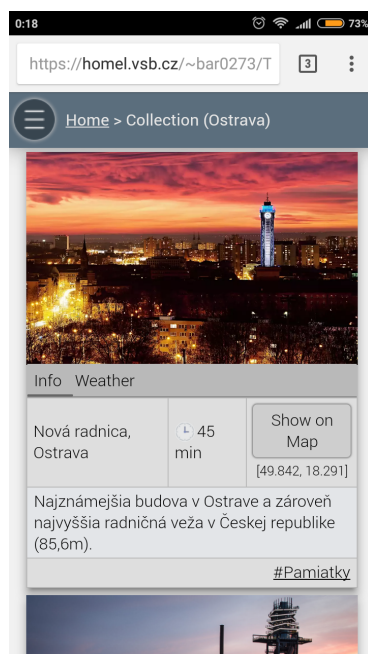
(c) Iphone 5SE

Obr. 13: Ilustrácia vyhľadávania trás medzi užívateľom definovanými bodmi na mape

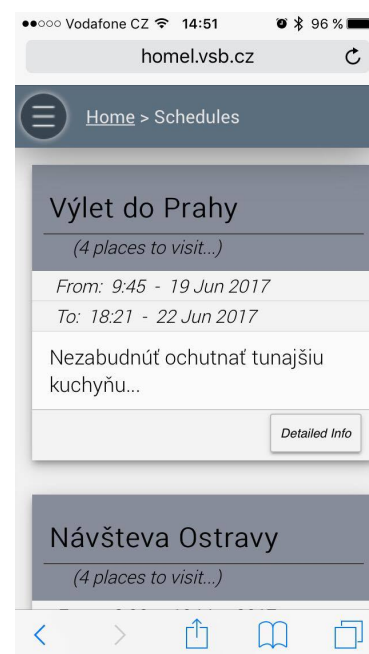
Testovanie pokračovalo prezeraním položiek z projektov. Túto časť testovania je možné vidieť na nasledujúcich obrázkoch 14a 14b 14c.



(a) Sony Xperia Z5 Compat



(b) Xiaomi Redmi 4A



(c) Iphone 5SE

Obr. 14: Ilustrácia manipulácie s obsahom projektov užívateľom

Pre užívateľov bola táto časť pomerne jednoduchá. Dokázali miesta zobrazovať pomocou všetkých triediacích kategórií. Zoznamy týchto položiek a položky samotné boli totiž dostatočne popísané a prehľadné. Nevýhodou bolo však to, že pre položky projektu a projekt celkovo nebol implementovaný žiaden spôsob, akým by mohol užívateľ dáta meniť, pridávať či mazať. Do budúcnosti sa navrhuje zahrnutie tejto funkcionality v rámci implementácie serverovej časti aplikácie.

6 Záver

V tejto práci sme úspešne vyvinuli mobilnú webovú aplikáciu, ktorá dokáže užívateľovi poskytnúť prvky pre orientáciu v priestore a plánovanie ciest. Podarilo sa nám vytvoriť interaktívnu mapu, ktorá využíva ako datový zdroj vektorové geodáta. Vyskúšali sme si, že jej vývoj je relatívne komplikovaným procesom. Dokázali sme pre ňu implementovať dynamické načítavanie obsahu ako aj ovládanie pomocou gest. Vyvinuli sme spôsob, ktorým si na nej užívateľ môže vyhľadávať optimalizované trasy medzi miestami. Vďaka využitiu indexovanej databázy sme tiež mape poskytli možnosť fungovania aj bez internetového pripojenia.

Vyriešených bolo ale i mnoho problémov spojených so správnym a efektívnym vykreslením obsahu mapy. Jednalo sa napríklad o rýchlosť vykreslenia, ktorú sme riešili cachovaním geodát, ale aj dynamické prispôbenie obsahu mapy k veľkosti prehliadača. Implementovali sme pre mapu vrstvu, ktorá správne vykresľuje názvy miest a to i pri dynamickom načítaní časti mapy.

Okrem toho sme pre užívateľa implementovali praktické vyhľadávanie miest a trás na mape a poskytli mu k nemu predpoveď počasia. Užívateľovi sme vytvorili základnú štruktúru projektov, vďaka ktorej je schopný si plánovať svoje cesty dopredu. Zároveň sme mu poskytli niekoľko možností zobrazenia a filtrácie obsahu jeho projektov. Poskytli sme mu tiež rýchlu navigáciu prehliadaného miesta kolekcie na mapu a nazad.

Celkovo sme sa naučili navrhovať webovú aplikáciu vzhľadom ku obmedzeniam mobilných platforiem. Testovaním sme zozbierali spätnú väzbu od užívateľov našej aplikácie, na základe ktorej sme navrhli jej možné vylepšenia.

V aplikácii je však stále dosť priestoru pre implementáciu ďalších prvkov a funkcionality. V rámci tejto práce boli pre účel aplikácie položené základy. V aplikácii by mohlo byť vyvinuté rozhranie pre manipuláciu s obsahom projektov a serverová časť aplikácie. V mape by sa mohlo implementovať rozmiestnenie štítkov. Mohla by sa tiež pridať funkcionality pre rezerváciu ubytovaní, vstupeniek atď. Práca by tak pre dosiahnutie svojho plného potenciálu mohla byť v budúcnosti ešte ďalej rozširovaná.

Literatura

- [1] Road Trip Route Planner, Map and Trip Guides. *Roadtrippers* [online]. [cit. 2017-03-29]. Dostupné z: <https://roadtrippers.com/>
- [2] Road Trip Route Planner, Map and Trip Guides. *About* [online]. [cit. 2017-03-27]. Dostupné z: <https://roadtrippers.com/about>
- [3] Najpokročilejšia GPS navigačná aplikácia - Sygic | Bringing life to maps. *Najpokročilejšia GPS navigačná aplikácia - Sygic* [online]. [cit. 2017-03-27]. Dostupné z: <http://www.sygic.com/sk/gps-navigation>
- [4] Sygic Travel. *Sygic Travel* [online]. [cit. 2017-03-29]. Dostupné z: <https://travel.sygic.com>
- [5] ORYSPAYEV, Dossay, Ramanathan SUGUMARAN a John DEGROOTE. *Geographic Information Systems: Concepts, Methodologies, Tools, and Applications (Development and Application of a Spreadsheet-Based Spatial Decision Support System (SDSS))*. Idea Group,U.S., 2012. ISBN 1466620382.
- [6] How People Use Their Devices - Think with Google. *How People Use Their Devices* [online]. [cit. 2017-03-26]. Dostupné z: <https://storage.googleapis.com/think/docs/twg-how-people-use-their-devices-2016.pdf>
- [7] Number of smartphone users worldwide 2014-2020 | Statista. *Number of smartphone users worldwide from 2014 to 2020 (in billions)* [online]. [cit. 2017-03-24]. Dostupné z: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- [8] GWI Device: Q3 2016 | Flagship Report Series. *GWI Device: Q3 2016 / Flagship Report Series* [online]. [cit. 2017-03-25]. Dostupné z: <http://insight.globalwebindex.net/device>
- [9] Mobile and tablet internet usage exceeds desktop for first time worldwide | StatCounter Global Stats. *Mobile and tablet internet usage exceeds desktop for first time worldwide* [online]. [cit. 2017-03-25]. Dostupné z: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>
- [10] Zenith forecasts 75% of internet use will be mobile in 2017 - Zenith. *Zenith forecasts 75% of internet use will be mobile in 2017* [online]. [cit. 2017-03-26]. Dostupné z: <https://www.zenithmedia.com/mobile-forecasts-75-internet-use-will-mobile-2017/>
- [11] ROZENTALS, Nathan. *Mastering TypeScript*. 23.04.2015. United Kingdom: Packt Publishing Limited, 2015. ISBN 1784399663.
- [12] jQuery. *What is jQuery?* [online]. [cit. 2017-03-27]. Dostupné z: <https://jquery.com/>

- [13] The Ultimate Guide to Mobile Website Development. *The Ultimate Guide to Mobile Website Development* [online]. [cit. 2017-04-09]. Dostupné z: <http://www.vervesearch.com/blog/the-ultimate-guide-to-developing-mobile-websites/>
- [14] Why Google Recommends Responsive Web Design | Social Media Today. *Why Google Recommends Responsive Web Design* [online]. [cit. 2017-04-09]. Dostupné z: <http://www.vervesearch.com/blog/the-ultimate-guide-to-developing-mobile-websites/>
- [15] Google Data. *Global Site Speed Overview: How Fast Are Websites Around The World?* [online]. [cit. 2017-03-31]. Dostupné z: <http://googledata.org/google-analytics/global-site-speed-overview-how-fast-are-websites-around-the-world/>
- [16] The need for mobile speed: How mobile latency impacts publisher revenue — Think with Google. *The need for mobile speed: How mobile latency impacts publisher revenue* [online]. [cit. 2017-03-31]. Dostupné z: <https://www.thinkwithgoogle.com/nordics/research-study/the-need-for-mobile-speed-how-mobile-latency-impacts-publisher-revenue/>
- [17] Mobile vs. Desktop: 10 Key Differences | ParadoxLabs. Mobile vs. Desktop: 10 Key Differences [online]. [cit. 2017-04-08]. Dostupné z: <https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/>
- [18] BUYYA, Rajkumar, Mukaddim PATHAN a Athena VAKALI, ed. *Content Delivery Networks*. Springer-Verlag Berlin Heidelberg, 2008. ISBN 978-3-540-77886-8.
- [19] Mobile User Experience: Limitations and Strengths. *Mobile Limitations and Strengths* [online]. [cit. 2017-04-09]. Dostupné z: <https://www.nngroup.com/articles/mobile-ux/>
- [20] The Thumb Zone: Designing For Mobile Users - Smashing Magazine. *The Thumb Zone: Designing For Mobile Users* [online]. [cit. 2017-03-31]. Dostupné z: <https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>
- [21] Mobile Auto-Suggest on Steroids: Tap-Ahead Design Pattern – Smashing Magazine. *Mobile Auto-Suggest on Steroids: Tap-Ahead Design Pattern* [online]. [cit. 2017-04-09]. Dostupné z: <https://www.smashingmagazine.com/2011/04/tap-ahead-design-pattern-mobile-auto-suggest-on-steroids/>
- [22] What Users Want Most from Mobile Sites Today – Think with Google. *What Users Want Most from Mobile Sites Today* [online]. [cit. 2017-04-02]. Dostupné z: <https://www.thinkwithgoogle.com/intl/en-gb/research-studies/what-users-want-most-from-mobile-sites-today.html>
- [23] Basic Patterns for Mobile Navigation. *Basic Patterns for Mobile Navigation: A Primer* [online]. [cit. 2017-04-09]. Dostupné z: <https://www.nngroup.com/articles/mobile-navigation-patterns/>

- [24] Reference Guide | User Interface | Settings. *Navigation Bar* [online]. [cit. 2017-04-08]. Dostupné z: <http://docs.miva.com/reference-guide/ui-settings>
- [25] Basic Patterns for Mobile Navigation. *Basic Patterns for Mobile Navigation* [online]. [cit. 2017-04-09]. Dostupné z: <http://babich.biz/basic-patterns-for-mobile-navigation/>
- [26] Techtopia. *Using Xcode Storyboards to create an iOS 5 iPhone Tab Bar Application* [online]. [cit. 2017-04-08]. Dostupné z: http://www.techotopia.com/index.php/Using_Xcode_Storyboards_to_create_an_iOS_5_iPhone_Tab_Bar_Application
- [27] Slide Menu – iOS | Android Made Easy. *How To Add a Slide-out Sidebar Menu in Your Apps* [online]. [cit. 2017-04-08]. Dostupné z: <http://agencenova.ch/wordpress/slide-menu-ios/>
- [28] Wayfinding For The Mobile Web – Smashing Magazine. *Hub and Spoke* [online]. [cit. 2017-04-08]. Dostupné z: <https://www.smashingmagazine.com/2014/10/wayfinding-for-the-mobile-web/>
- [29] Promise Object (JavaScript) | Microsoft Docs. *Promise Object (JavaScript)* [online]. [cit. 2017-04-21]. Dostupné z: <https://docs.microsoft.com/en-us/scripting/javascript/reference/promise-object-javascript>
- [30] Optimized route - Mapzen Mobility. *Mapzen Mobility* [online]. [cit. 2017-04-27]. Dostupné z: <https://mapzen.com/documentation/mobility/>
- [31] What is RESTful API? - Definition from WhatIs.com. *RESTful API* [online]. [cit. 2017-04-26]. Dostupné z: <http://searchcloudstorage.techtarget.com/definition/RESTful-API>
- [32] Mapzen Search. *Mapzen Search* [online]. [cit. 2017-04-26]. Dostupné z: <https://mapzen.com/documentation/search/>
- [33] Vector Tile Service. *Vector Tile Service* [online]. [cit. 2017-04-26]. Dostupné z: <https://mapzen.com/documentation/vector-tiles/>
- [34] Slick: *slick - the last carousel you'll ever need* [online]. [cit. 2017-07-17]. Dostupné z: <https://kenwheeler.github.io/slick/>
- [35] Web Storage API - Web APIs | MDN. *Web Storage API* [online]. [cit. 2017-04-24]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
- [36] Can I use.. Support tables for HTML5, CSS3, etc. *Can I use* [online]. [cit. 2017-04-27]. Dostupné z: <https://caniuse.com/>

A Obsah CD

1. TravelGuideApplication (Visual Studio Solution)
2. readme.txt
3. Tento dokument vo formáte .pdf